New optimization techniques for point feature and general curve feature based SLAM

Minjie Liu

Submitted in fulfilment of the requirement for the degree of doctor of philosophy

Mechatronics and Intelligent Systems Group
Center for Autonomous Systems
The Faculty of Engineering and Information Technology
The University of Technology, Sydney

Student: : Minjie Liu

Supervisor : Dr Shoudong Huang
Co-Supervisor : Prof Gamini Dissanayake

Certificate

I, Minjie Liu, declare that this thesis titled, “New optimization techniques for point feature and general curve feature based SLAM" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

Acknowledgements

I would like to express an enormous amount of gratitude to all the people who have supported and encouraged me during the past three and a half years of my candidature.

Firstly I would like to thank my supervisors, Dr Shoudong Huang and Professor Gamini Dissanayake for their invaluable guidance and supervision, and for providing me this opportunity.

Thanks goes to Dr Heng Wang, Mr Liang Zhao, Dr, Zhan Wang, Dr Alen Alempijevic and Mr Gibson Hu who have given useful information, suggestions, criticisms and helped my research. I would also thank Lei Shi, Shifeng Wang, Shuai Yuan and Chuan Zhao for their valued friendship.

Most importantly, I would like to thank my family. For my parents, thank you for your love and care. Without their support and understanding this would not have been possible. For my lovely wife Chen Zheng, thank you for your love, support and encouragement throughout all these years.

Contents

List of Tables

List of Figures

Abstract

This doctoral thesis deals with the feature based Simultaneous Localization and Mapping (SLAM) problem. SLAM as defined in this thesis is the process of concurrently building up a map of the environment and using this map to obtain improved estimates of the location of the robot. In feature based SLAM, the robot relies on its ability to extract useful navigation information from the data returned by its sensors. The robot typically starts at an unknown location without priori knowledge of feature locations. From relative observations of features and relative pose measurements, estimates of entire robot trajectory and feature locations can be derived. Thus, the solution to SLAM problem enables an autonomous vehicle navigates in a unknown environment autonomously. The advantage of eliminating the need for artificial infrastructures or a priori topological knowledge of the environment makes SLAM problem one of the hot research topics in the robotics literature. Solution to the SLAM problem would be of inestimable value in a range of applications such as exploration, surveillance, transportation, mining etc.

The critical problems for feature based SLAM implementations are as follows: 1) Because SLAM problems are high dimensional, nonlinear and non-convex, when solving SLAM problems, robust optimization techniques are required. 2) When the environment is complex and unstructured, appropriate parametrization method is required to represent environments with minimum information loss. 3) As robot navigates in the environment, the information acquired by the onboard sensor increases. It is essential to develop computationally tractable SLAM algorithms especially for general curve features.

This thesis presents the following contributions to feature based SLAM. First, a convex optimization based approach for point feature SLAM problems is developed. Using the proposed method, a unique solution can be obtained without any initial state estimates. It will be shown that, the unique SDP solution obtained from the proposed method is very close to the true solution to the SLAM problem. Second, a general curve feature based SLAM formulation is presented. Instead of scattered points, in this formulation, the envi-

ronment is represented by a number of continuous curves. Using the new formulation, all the available information from the sensor is utilized in the optimization process. Third, method for converting curve feature to point feature is presented. Using the conversion method, the curve feature SLAM problem can be transferred to point feature SLAM problem and can be solved by the convex optimization based approach.

1 Introduction

Reliable localization is the key concept of any autonomous robotic system. Autonomous mobile robot localization in previously unexplored environments requires the robot to incrementally construct a map of its surroundings and using this map to obtain improved estimates of the location of the robot. This problem is called Simultaneous Localization and Mapping (SLAM) problem.

One of the most popular approach to the SLAM problem is the feature based SLAM. In feature based SLAM, features from the raw sensor measurements have been firstly extracted and the raw measurements have been transferred to the relative robot-to-feature observations. Estimation technique is then applied on this information and the relative robot pose measurement to get the robot trajectory and the feature locations.

This thesis is concerned with the convex optimization based approach to feature based SLAM. The major contributions of this thesis arise from the new formulation to the point feature SLAM, on which convex optimization based approach can be applied. Because the use of the convex optimization technique, unique solution can be obtained without any initial state estimates. To utilize more information from raw sensor measurement, a general curve feature based SLAM formulation is proposed. By introducing a new observation model, curve features are transferred into point features and the proposed convex optimization based approach can be used to solve curve feature based SLAM problem.

1.1 Background and Motivation

Autonomous mobile robots play an important role in many areas of the modern society. From service robots such as vacuum-cleaners and garden robots to industrial robots such as Automated Guided Vehicle (AGV), robots effectively and efficiently performs human-

like, labor-intensive tasks. The success of deploying robots in modern society has led to significant innovation in the research and development of robotic systems. As unmanned vehicles allow often dangerous tasks to be performed from remote locations in a range of application domains such as mining, defence and subsea exploration, mobile robotics becomes one of the key research area in robotic systems.

There are several important facets of autonomous vehicles and one of them is reliable localization. Over the past decades, robotics researchers have developed reliable solutions to the problem of precision location estimation within known environments. Typically, artificial infrastructures with known locations have been arranged in the environment and location of the vehicle is computed by integrating information from sensors such as sonar, lasers and radars mounted on the vehicle that observe the artificial infrastructures with information from sensors that monitor the behavior of the vehicle, such as encoders and inertial measurement units.

However, most of the time installed infrastructure at known locations is not available and external sources of information such as GPS is not present. In contrast with the localization problem, SLAM considers the problem which uses only relative measurements to build up a map of the environment and at the same time to obtain improved estimates of the location of the vehicle.

After the introduction of the SLAM problem, it soon became one of the most active research areas in mobile robotics and many SLAM algorithms have been developed in the past decade. Generally speaking, the SLAM problem is considered as a nonlinear estimation/optimization problem. Solving such a problem is not a simple task. As robot navigates in large and complicated environment, the amount of computational effort required grows quickly.

Now, it is well recognized that SLAM problem has a sparse structure. This significantly reduce the computational cost for solving this high dimensional nonlinear optimization problem. As such, a number of efficient optimization based SLAM algorithms have emerged in the last few years (e.g. [18][43] and the references therein).

Recent research in optimization based SLAM has demonstrated some surprising results. It is evident from [29][63] that the pose SLAM problem can converge to the correct solution most of the time even if it starts from a poor initial guess, provided that the covariance matrix of the relative pose information is near spherical.

Some recent work on feature based SLAM [33] also demonstrates a similar phenomenon. A solution that is very close to the published results for the popular Victoria Park dataset [30] can be obtained from a simple Gauss-Newton algorithm, using an odometry based initial guess and setting all the covariance matrices to identity. More interestingly, even with random initial guesses to the 6898 vehicle poses and the 299 feature positions, 80% of the time the same result can be obtained, contrary to the obvious expectation that the algorithm will converge to a local minimum. In case of the DLR-Spatial Cognition dataset [44] which is another popular dataset used by SLAM researchers, the same strategy results in a solution close to the ground truth if a zero vector is used as the initial guess while in contrast a random initial guess leads to a local minimum.

The above phenomenon shows that SLAM optimization problem is a very special nonlinear optimization problem and possibly it is close to a convex optimization problem. Work reported in this thesis is motivated by this phenomenon and tried to explored the hidden convexity of the SLAM problem. It is well known that any local minimum of a convex optimization problem is the global minimum of the problem. Thus solving the SLAM problem using convex optimization based method is potentially a robust strategy for obtaining the true optimal solution to SLAM problem.

1.2 Contributions

This thesis focuses on the convex optimization based SLAM problem. The principal contributions of this thesis arise from the new formulation of the SLAM problem which explores hidden convex structure of the problem. The contributions made are:

- A convex optimization based approach to the point feature based SLAM. This method first transfers the SLAM problem to a quadratically constrained quadratic programming (QCQP) problem by using a new parametrization of the robot orientation and some approximations on the structure of the uncertainty of measurements. The resulting QCQP, which is not convex, is then relaxed using semi-definite relaxation (SDR) to obtain a semi-definite programming (SDP) problem which is convex. The unique solution to the SDP can be computed even when a suitable initial guess is not available. Using computer simulation and experimental data, it is shown that a candidate solution to the SLAM problem extracted from the unique

SDP solution is very close to the true solution to the SLAM problem.

- A new formulation for curve feature SLAM is proposed. In this formulation, the environment is represented by a number of continuous curves. It has been observed that, when using the new formulation, the objective function to be optimized becomes discontinuous and can not be solved by standard gradient based approaches. It is proposed to solve this problem by using a Genetic Algorithm (GA) and Iterative Closest Point (ICP) based approach. However, although this approach is able to handle discontinuities, the computation cost involved is too high.
- A new observation model for B-Spline SLAM is proposed. Curve feature in the environment are represented by B-Splines. Using the new observation model, the curve feature SLAM problem can be transferred to point feature SLAM problem. This enables more information from the raw sensor measurements to be utilized. Using the new observation model, the proposed convex optimization based approach can also be applied to B-Spline SLAM.

1.3 Publications

Following is a list of publications towards the work presented in this thesis:

- "A new observation model for B-Spline SLAM," M. Liu, S. Huang and G. Dissanayake. *Australasian Conference on Robotics and Automation (ACRA)*, Sydney, Australia, 2009.
- "Towards a consistent SLAM algorithm using B-Splines to represent environments," M. Liu, S. Huang, G. Dissanayake and S. Kodagoda, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2065-2070, 2010.
- "Feature based SLAM using laser sensor data with maximized information usage," M. Liu, S. Huang and G. Dissanayake, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1811-1816, 2011.
- "A convex optimization based approach for point feature SLAM problems," M. Liu, H. Wang, S. Huang, and G. Dissanayake *Submitted to IEEE Transactions on Robotics 2011.*

- "A convex optimization based approach for pose SLAM problems," M. Liu, S. Huang, G. Dissanayake and H. Wang *Accepted by IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* 2012.

1.4 Thesis Overview

The remainder of this thesis is organized as follows.

Chapter 2 reviews some of the existing approaches for SLAM problem. In particular, optimization based approaches for feature based SLAM are discussed. Some of the existing feature representation methods are also presented.

Chapter 3 presents a convex optimization based approach to the SLAM problem. A new representation for robot orientation is introduced such that the point feature SLAM problem can be formulated as a quadratic constrained quadratic programming (QCQP) optimization problem which can then be relaxed to a convex optimization problem.

Chapter 4 evaluates the proposed convex optimization approach for point feature SLAM. A number of comparison methods such as consistency of the algorithm and value of objection function are used. Results have shown that the results derived from the proposed approach is very close to the results derived from least squares method.

Chapter 5 proposes a curve feature SLAM formulation. Using this formulation all the raw measurements have been utilized in the objective function. However, it is observed that the objective function for the new formulation is discontinuous. Because of this, local optimization based approach can not be applied. Thus a Genetic Algorithm (GA) based method is proposed.

Chapter 6 proposes a new observation model for B-Spline based SLAM. The use of B-Splines allows environment to be represented without having to extract specific geometric features such as lines or points. Using the new observation model, the curve feature based SLAM problem has been transferred to point feature SLAM problem which can be solved using both nonlinear least squares and the proposed convex optimization approach.

Chapter 7 presents conclusions and provides directions for future research of the proposed approach.

2 Simultaneous Localization and Mapping

2.1 Introduction

Since the introduction of the Simultaneous Localization and Mapping (SLAM) problem, the problem has attracted immense attention from the mobile robotics community. When a mobile robot starts at an unknown location with no a priori knowledge of the environment, solution to the SLAM problem enables the robot concurrently building a map of the environment and using this map to obtain estimates of its location.

Probably the first approach to SLAM problem is the feature based SLAM introduced by Smith, Self and Cheeseman [72][73] that built on previous work by Ayache and Faugeras [3] and Chatila and Laumond [16]. In feature based SLAM approach, features have been firstly extracted from the raw measurements returned from the onboard sensors. From the relative robot-to-features observations and relative relative robot pose measurement (also known as odometry information), the estimate of robot trajectory and the estimate of feature locations can be computed.

Early work on feature based SLAM focuses on point-features [20]. When the point feature SLAM problem has been first introduced, the Extended Kalman Filter (EKF) has been proposed as the estimator. In EKF based SLAM approaches the information gathered by the robot can be fused to yield bounded estimates [81] of robot and feature locations in a recursive fashion.

After years of development for SLAM, there are many advancement for feature based SLAM algorithms. Firstly, the point feature based SLAM problem has been formulated as an optimization problem. Instead of estimating only the current robot pose as in EKF, the optimization approach estimates the entire robot trajectory. Results show that

SLAM algorithms using optimization techniques can improve the consistency significantly as Jacobians are re-evaluated over different times [19][36]. Secondly, the sparseness of the information matrix in different point feature SLAM formulations is now well understood and exploited thoroughly [1][19][35]. Hence the problem can be solved efficiently for scenarios consisting of a few thousand of robot to feature observations. Further more, more complex features have been used to represent the environment such that more information can be utilized in the estimation process and more detailed description of the environment can be achieved.

Different from the EKF and optimization based approach to the SLAM problem, another popular SLAM approach is the particle filter based SLAM [56][26]. It uses a Rao-Blackwellized particle filter to sample robot poses and track the position of a fixed number of predetermined features using EKF (the feature positions are conditionally independent given the robot poses). This method mitigates some of the challenges in mapping at the expense of some challenges in feature selection and identification. For particle filter based SLAM, the more the number of particles used, the less the extent of the inconsistency.

Another mainstream SLAM approach is the pose SLAM (also known as Graph SLAM) [29][48][63]. In pose SLAM, relative pose information between consecutive robot poses are computed and then an optimization is performed to smooth the whole robot trajectory.

While all these approaches to the problem have their own particular strengths, this thesis will mainly focus on the feature based SLAM problem using optimization techniques.

This chapter briefly introduce some of the existing feature based SLAM algorithms used for generating robot and feature position estimates based on observations taken relative to the position of the robot for 2D case. Section 2.2 begins by introducing the system states for least squares SLAM formulation. Section 2.3 describes the motion and observation model used in point feature SLAM. Section 2.4 presents the point feature SLAM algorithm using non-linear least squares as estimator. In Section 2.5, least squares estimation method is discussed. Section 2.6 reviews some of the least squares based algorithm for feature based SLAM. Section 2.7 discusses some of the complex feature representation methods and Section 2.8 summarizes this chapter.

2.2 Systems State

In feature based SLAM, the environment is represented by the absolute position of the features. For example, in point feature based SLAM, the map is represented by a set of points. Considering the 2D case, the $j-$th point feature X_{f_j} is represented by its position

$$X_{f_j} = \begin{bmatrix} x_{f_j} \\ y_{f_j} \end{bmatrix}.$$

And the $i-$th robot pose X_{r_i} is represented by

$$X_{r_i} = \begin{bmatrix} x_{r_i} \\ y_{r_i} \\ \phi_{r_i} \end{bmatrix}$$

where $(x_{r_i},\ y_{r_i})$ is the absolute robot position and ϕ_{r_i} is the absolute robot orientation. The initial robot pose has been set as the origin i.e. $X_{r_0} = [0, 0, 0]^T$.

Suppose the robot moved p times and there are altogether N features have been observed, the system state for nonlinear least squares point feature SLAM formulation is

$$V = [X_{r_1}^T, \ldots, X_{r_p}^T, X_{f_1}^T, \ldots, X_{f_N}^T]^T. \tag{2.1}$$

2.3 The Motion and Observation Model

In feature based SLAM, measurements are in the relative coordinate system. In order to get feature and robot pose estimate in global coordinate system, models are needed to map between the sensed data (e.g. laser, cameras, odometry) and the states of interest (e.g. robot poses, stationary feature poses). There are two models in feature based SLAM, observation model and motion model. Using point feature SLAM as an example, we discuss the two models in this section.

2.3.1 Observation model

Suppose at robot pose i, the robot observes point feature X_{f_j} and assume the robot-to-feature observation is the relative position measurement of point feature f_j with respect to robot pose X_{r_i}

$$z_j^i = \begin{bmatrix} z_{x_j^i} \\ z_{y_j^i} \end{bmatrix},$$

the observation model can be written as

$$z_j^i = H^{z_j^i}(X_{r_i}, X_{f_j}) + \sigma_{z_j^i} \tag{2.2}$$

where $\sigma_{z_j^i}$ is the noise on the observation z_j^i. It is assumed to be Gaussian with zero mean and covariance matrix $P_{z_j^i}$. $H^{z_j^i}$ is the observation function given by

$$\begin{aligned} H^{z_j^i}(X_{r_i}, X_{f_j}) &= \begin{bmatrix} \cos(\phi_{r_i})(x_{f_j} - x_{r_i}) + \sin(\phi_{r_i})(y_{f_j} - y_{r_i}) \\ -\sin(\phi_{r_i})(x_{f_j} - x_{r_i}) + \cos(\phi_{r_i})(y_{f_j} - y_{r_i}) \end{bmatrix} \\ &= \begin{bmatrix} \cos(\phi_{r_i}) & \sin(\phi_{r_i}) \\ -\sin(\phi_{r_i}) & \cos(\phi_{r_i}) \end{bmatrix} \begin{bmatrix} x_{f_j} - x_{r_i} \\ y_{f_j} - y_{r_i} \end{bmatrix}. \end{aligned} \tag{2.3}$$

The Jacobian matrix of $H^{z_j^i}$ with respect to robot pose X_{r_i} is

$$\frac{\partial H^{z_j^i}}{\partial X_{r_i}} = \begin{bmatrix} -\cos(\phi_{r_i}) & -\sin(\phi_{r_i}) & -\sin(\phi_{r_i})(x_{f_j} - x_{r_i}) + \cos(\phi_{r_i})(y_{f_j} - y_{r_i}) \\ \sin(\phi_{r_i}) & -\cos(\phi_{r_i}) & -\cos(\phi_{r_i})(x_{f_j} - x_{r_i}) - \sin(\phi_{r_i})(y_{f_j} - y_{r_i}) \end{bmatrix} \tag{2.4}$$

and the Jacobian matrix of $H^{z_j^i}$ with respect to feature position X_{f_j} is

$$\frac{\partial H^{z_j^i}}{\partial X_{f_j}} = \begin{bmatrix} \cos(\phi_{r_i}) & \sin(\phi_{r_i}) \\ -\sin(\phi_{r_i}) & \cos(\phi_{r_i}) \end{bmatrix}. \tag{2.5}$$

2.3.2 Motion model

Suppose the relative robot pose measurement (also known as odometry information) of robot pose $X_{r_{i+1}}$ with respect to robot pose X_{r_i} is in the following form

$$o_i = \begin{bmatrix} o_{x_i} \\ o_{y_i} \\ o_{\phi_i} \end{bmatrix}, \tag{2.6}$$

where $(o_{x_i},\ o_{y_i})$ is the relative robot position measurement and o_{ϕ_i} is the relative robot orientation measurement. The motion model is

$$o_i = H^{o_i}(X_{r_i}, X_{r_{i+1}}) + \sigma_{o_i}. \tag{2.7}$$

Here σ_{o_i} is the noise on the odometry information o_i. It is assumed to be Gaussian with zero mean and covariance matrix P_{o_i}. $H^{o_i}(X_{r_i}, X_{r_{i+1}})$ is the odometry function given by

$$H^{o_i}(X_{r_i}, X_{r_{i+1}}) = \begin{bmatrix} \cos(\phi_{r_i})(x_{r_{i+1}} - x_{r_i}) + \sin(\phi_{r_i})(y_{r_{i+1}} - y_{r_i}) \\ -\sin(\phi_{r_i})(x_{r_{i+1}} - x_{r_i}) + \cos(\phi_{r_i})(y_{r_{i+1}} - y_{r_i}) \\ \phi_{r_{i+1}} - \phi_{r_i} \end{bmatrix}. \tag{2.8}$$

The Jacobian matrix of $H^{o_i}(X_{r_i}, X_{r_{i+1}})$ with respect to X_{r_i} is

$$\frac{\partial H^{o_i}}{\partial X_{r_i}} = \begin{bmatrix} -\cos(\phi_{r_i}) & -\sin(\phi_{r_i}) & -\sin(\phi_{r_i})(x_{r_{i+1}} - x_{r_i}) + \cos(\phi_{r_i})(y_{r_{i+1}} - y_{r_i}) \\ \sin(\phi_{r_i}) & -\cos(\phi_{r_i}) & -\cos(\phi_{r_i})(x_{r_{i+1}} - x_{r_i}) - \sin(\phi_{r_i})(y_{r_{i+1}} - y_{r_i}) \\ 0 & 0 & -1 \end{bmatrix} \tag{2.9}$$

and the Jacobian matrix of $H^{o_i}(X_{r_i}, X_{r_{i+1}})$ with respect to $X_{r_{i+1}}$ is

$$\frac{\partial H^{o_i}}{\partial X_{r_{i+1}}} = \begin{bmatrix} \cos(\phi_{r_i}) & \sin(\phi_{r_i}) & 0 \\ -\sin(\phi_{r_i}) & \cos(\phi_{r_i}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.10}$$

2.4 Least Squares Formulation for Point Feature SLAM

Given the observation and motion models described in the previous section, the point feature SLAM problem is to derive the system state which best fits all the available information. This can be formulated as a least squares problem.

There are two scenarios for point feature SLAM: 1) odometry information is unavailable; 2) odometry is available. Suppose there are N point features $f_1, ..., f_N$ that are observed from a sequence of $p+1$ robot poses $r_0, r_1, \ldots, r_p$. Assuming that at robot pose r_i, the robot observes k_i features out of total N features, the point feature SLAM least squares formulation for these two scenarios are discussed.

2.4.1 When odometry information is unavailable

The point feature SLAM problem in this case aims to find the state vector V defined in (2.1), which contains all the robot poses and feature locations to best fit all the observations. This can be achieved by minimizing the following least squares objective function:

$$F_z(V) = \sum_{i=0}^{p} \sum_{j=1}^{k_i} (z_j^i - H^{z_j^i}(V))^T P_{z_j^i}^{-1} (z_j^i - H^{z_j^i}(V)) \tag{2.11}$$

Fig. 2.1 shows an example of the case when odometry information is unavailable with $N = 6$ and $p = 4$.

Similar to D-SLAM [80], when odometry is unavailable, common features need to be observed from different robot poses such that the robot poses can be estimated. When the SLAM problem is formulated as the above least squares optimization problem, only one instance when less than two previously seen features are observed is allowed (e.g. pose r_2 in Fig 2.1). If this condition is not satisfied, the information available is not enough to uniquely determine the robot poses.

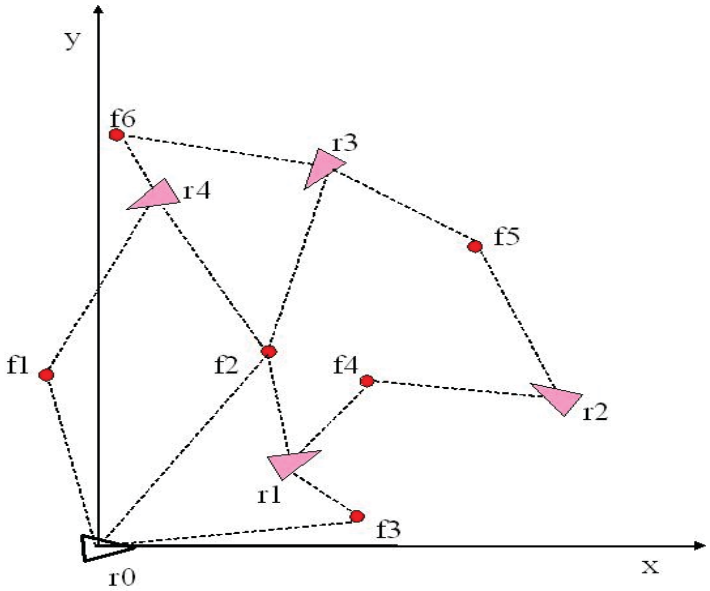


Figure 2.1: The SLAM problem with 5 poses, 6 features and 14 observations (odometry unavailable case).

2.4.2 When odometry information is available

The point feature SLAM problem in this case is to use the odomety and observation information to estimate all the robot poses and all the feature positions. This can be achieved by minimizing the least squares objective function:

$$\begin{aligned} F_z(V) + F_o(V) = & \sum_{i=0}^{p} \sum_{j=1}^{k_i} (z_j^i - H^{z_j^i}(V))^T P_{z_j^i}^{-1} (z_j^i - H^{z_j^i}(V)) \\ & + \sum_{i=0}^{p-1} (o_i - H^{o_i}(V))^T P_{o_i}^{-1} (o_i - H^{o_i}(V)) \end{aligned} \tag{2.12}$$

Fig. 2.2 shows an example of this scenario with $N = 3$ and $p = 4$. When odometry is available, the information is always enough to determine the variables no matter how many features are observed at each time step.

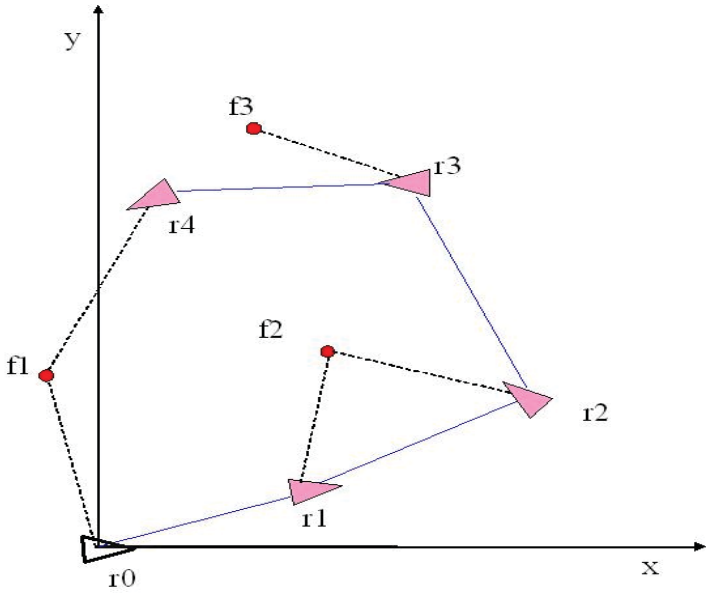


Figure 2.2: The SLAM problem with 5 poses, 3 features and 5 observations (odometry available case).

2.5 Least Squares Estimation Method

When formulating feature based SLAM as a least squares optimization problem, the SLAM problem can be solved by least squares technique. The least squares technique is a frequently used approach to solving over-determined or inexactly specified systems of equations in an approximate sense. Instead of solving the equations exactly, least squares aims to minimize the sum of the squares of the residuals.

This section briefly reviews the least squares estimation method. Firstly, the linear least squares method is described and then the two popular non-linear least squares methods (Gauss-Newton and Levenberg-Marquardt) are discussed.

2.5.1 Least squares problem

In a general least squares problem, the objective is to adjust the variables of a model function to best fit a data set. Assuming there are m observations

$$Z = [Z_1, Z_2, \ldots, Z_m]^T$$

and the model function has the form of

$$F_i(V),\ i = 1, \ldots, m$$

least squares aims to find

$$V = [V_1, V_2, \ldots, V_n],\ n < m$$

which minimizes the following objective function

$$\sum_{i=1}^{m} \|F_i(V) - Z_i\|^2 \tag{2.13}$$

Depending on the problem, the model function can be in various forms.

2.5.2 Linear least squares

Consider a multi-dimensional case and the model function comprises a linear combination of the parameter i.e.

$$F_i(V) = \sum_{j=1}^{n} Y_{i,j} V_j \tag{2.14}$$

where

$$Y_{i,j},\ i = 1, \ldots, m,\ j = 1, \ldots, n$$

are known coefficients. Least squares problem (2.13) becomes minimizing

$$\sum_{i=1}^{m} \| \sum_{j=1}^{n} Y_{i,j} V_j - Z_i\|^2 \tag{2.15}$$

Rewrite (2.15) in matrix form

$$\min_{V} \|YV - Z\|^2 \tag{2.16}$$

where

$$Y = \begin{bmatrix} Y_{11} & \dots & Y_{1n} \\ \vdots & \ddots & \vdots \\ Y_{m1} & \dots & Y_{mn} \end{bmatrix}, \; V = \begin{bmatrix} V_1 \\ \vdots \\ V_n \end{bmatrix} \text{ and } Z = \begin{bmatrix} Z_1 \\ \vdots \\ Z_m \end{bmatrix}$$

expanding the above expression, (2.16) becomes

$$\begin{aligned} ||Z - YV||^2 &= (Z - YV)^T (Z - YV) \\ &= Z^T Z - Z^T YV - V^T Y^T Z + V^T Y^T YV. \end{aligned} \tag{2.17}$$

The objective can then be achieved by finding V that makes the following Jacobian function to 0

$$\frac{d}{dV}(Z^T Z - Z^T YV - V^T Y^T Z + V^T Y^T YV) = 0. \tag{2.18}$$

The solution for this is

$$V = (Y^T Y)^{-1} Y^T Z. \tag{2.19}$$

2.5.3 Nonlinear least squares

In the nonlinear least squares case, the model function is nonlinear in terms of the parameter V. Unlike the linear least squares case, there is no closed-form solution. Therefore, the principle for the non-linear least squares estimation is to use numerical method to find the solution iteratively. Here, two of the most commonly used nonlinear least squares techniques are introduced.

Gauss-Newton method

Gauss-Newton is an algorithm to solve unconstrained least squares problem. In Gauss-Newton method, the nonlinear system $F(V)$ is represented by its Taylor expansion

$$F(V) = F(V_0) + \frac{F'(V_0)}{1!}(V - V_0) + \frac{(V - V_0)^T F''(V_0)(V - V_0)}{2!} + \dots \tag{2.20}$$

and is approximated by

$$F(V) \approx F(V_0) + \frac{F'(V_0)}{1!}(V - V_0). \tag{2.21}$$

When V is near V_0 the minimization problem can be approximated by

$$||Z - F(V_0) + F'(V_0)V_0 - F'(V_0)V||^2. \tag{2.22}$$

From (2.19), given an initial value V_0, the one step improved approximate solution for nonlinear least squares problem is

$$V_1 = (F'(V_0)^T F'(V_0))^{-1} F'(V_0)^T [Z - F(V_0) + F'(V_0)V_0]. \tag{2.23}$$

Levenberg-Marquardt method

Similar to the Gauss-Newton method, the Levenberg-Marquardt method is another method for solving non-linear least squares problems. Levenberg-Marquardt method calculates an increment ΔV which minimizes the objective function. Similar to (2.21), in Levenberg-Marquardt method $F(V)$ is approximated by

$$F(V) \approx F(V_0) + \frac{F'(V_0)}{1!}\Delta V \tag{2.24}$$

where

$$\Delta V = V - V_0 \tag{2.25}$$

and (2.13) is approximated by

$$||Z - F(V_0) - F'(V_0)\Delta V||^2. \tag{2.26}$$

Thus, the objective is to find ΔV to minimize (2.26). Define

$$\xi = Z - F(V_0),$$

the objective can be achieved by the following equation

$$(F'(V_0))^T F'(V_0)\Delta V = (F'(V_0))^T \xi. \tag{2.27}$$

Notice that (2.27) is equivalent to (2.23) in Gauss-Newton.

Different from Gauss-Newton, in Levenberg-Marquardt a damping version of (2.27) is introduced

$$((F'(V_0))^T F'(V_0) + \lambda I)\Delta V = (F'(V_0))^T \xi. \tag{2.28}$$

where λ is a non-negative damping factor. This update rule is used as follows. At a large distance from the function minimum, large λ value is utilized to provide steady and convergent progress toward the solution. As the solution approaches the minimum, λ is adaptively decreased to reduce the influence of gradient descent. An efficient updating strategy for the selecting the damping factor has been described in [59].

2.6 SLAM Algorithms Based on Least Squares

Recently, there are a number of algorithms emerged to reduce the computational cost for least square based SLAM utilizing the special structure of the problem. In this section, we provide a brief review on some of the algorithms.

2.6.1 Square root SAM

In *Square Root Smoothing and Mapping* (SAM) [19], the point feature SLAM nonlinear least squares formulation is solved by using the QR matrix factorization. Comparing with the nonlinear least squares method (2.23), SAM avoids calculating $F'(V_0)^T F'(V_0)$ which reduces the computation cost significantly.

Letting

$$b = F(V_0) - Z - F'(V_0)V_0,$$

(2.22) can be written as

$$||F'(V_0)V - b||^2 \tag{2.29}$$

Suppose the QR factorization of matrix $F'(V_0)$ is

$$F'(V_0) = Q \begin{bmatrix} R \\ 0 \end{bmatrix} \tag{2.30}$$

where R is a upper triangular matrix and Q is a orthogonal matrix. Applying the factorization, (2.29) can be written as

$$||F'(V_0)V - b||^2 = ||Q \begin{bmatrix} R \\ 0 \end{bmatrix} V - b||^2. \tag{2.31}$$

Multiply the above expression by Q^T, the objective function can be written as

$$||Q^T Q \begin{bmatrix} R \\ 0 \end{bmatrix} V - Q^T b||^2 \tag{2.32}$$

Since Q is an orthogonal matrix (2.32) can be simplified as

$$|| \begin{bmatrix} R \\ 0 \end{bmatrix} V - Q^T b||^2. \tag{2.33}$$

Define $[d, e]^T = Q^T b$, (2.32) becomes

$$\| \begin{bmatrix} R \\ 0 \end{bmatrix} V - \begin{bmatrix} d \\ e \end{bmatrix} \|^2 = ||RV - d||^2 + ||e||^2. \tag{2.34}$$

Since $||e||^2$ is a constant value, and R is a upper triangular matrix, solution to (2.22) can be derived by solving the following linear equation

$$RV = d. \tag{2.35}$$

Because R is a upper triangular matrix, V can be solved by back substitution.

2.6.2 Sparse bundle adjustment

Sparse Bundle Adjustment (SBA) is another popular method for solving the point feature SLAM problem in vision [76][1][17]. This method is based on the Levenberg-Marquardt method.

In SBA, the information matrix $(F'(V_0))^T F'(V_0)$ still need to be evaluated. Let

$$H(V_0) = (F'(V_0))^T F'(V_0)$$

and since the order for V is robot poses first and features second, $H(V_0)$ can be partitioned as

$$H(V_0) = \begin{bmatrix} H(V_0)_{rr} & H(V_0)_{rf} \\ H(V_0)_{fr} & H(V_0)_{ff} \end{bmatrix} \tag{2.36}$$

and $F'(V_0)$ can be partitioned as

$$F'(V_0) = \begin{bmatrix} F'(V_0)_r & F'(V_0)_f \end{bmatrix}^T. \tag{2.37}$$

From (2.36) and (2.37), (2.27) can be rewritten as

$$\begin{bmatrix} H(V_0)_{rr} & H(V_0)_{rf} \\ H(V_0)_{fr} & H(V_0)_{ff} \end{bmatrix} \begin{bmatrix} \Delta r \\ \Delta f \end{bmatrix} = \begin{bmatrix} F'(V_0)_r \xi_r \\ F'(V_0)_f \xi_f \end{bmatrix}. \tag{2.38}$$

When multiplying (2.38) by the block matrix

$$\begin{bmatrix} I & -H(V_0)_{rr} H(V_0)_{ff}^{-1} \\ 0 & I \end{bmatrix} \tag{2.39}$$

(2.38) becomes

$$\begin{bmatrix} H(V_0)_{rr} - H(V_0)_{rf}H(V_0)_{ff}^{-1}H(V_0)_{fr} & 0 \\ H(V_0)_{fr} & H(V_0)_{ff} \end{bmatrix} \begin{bmatrix} \Delta r \\ \Delta f \end{bmatrix} = \begin{bmatrix} F'(V_0)_r\xi_r - H(V_0)_{rr}H(V_0)_{ff}^{-1}F'(V_0)_f\xi_f \\ F'(V_0)_f\xi_f \end{bmatrix}. \tag{2.40}$$

Thus, the pose updating vector Δr can be first calculated using

$$(H(V_0)_{rr} - H(V_0)_{rf}H(V_0)_{ff}^{-1}H(V_0)_{fr})\Delta r = F'(V_0)_r\xi_r - H(V_0)_{rr}H(V_0)_{ff}^{-1}F'(V_0)_f\xi_f \tag{2.41}$$

and Δf can be computed by back substituting

$$H(V_0)_{fr}\Delta r + H(V_0)_{ff}\Delta f = F'(V_0)_f\xi_f \tag{2.42}$$

Generally speaking, the number of features in feature based SLAM problem is significantly larger than the number of robot poses. Thus, by calculating the pose updating vector Δr first and back substituting Δr to get Δf, the computation cost can be significantly reduced.

2.6.3 Submap based approach

Different with the batch methods which estimates the entire set of robot poses and features in the environment described in Section 2.6.1 and Section 2.6.2, another common approach to reduce the computational cost is the *submap* [21][11][24][61][35][66] based approach.

The basic idea of submap method is to first build a sequence of small sized local submaps and then combine the local submaps into a large-scale global map. In the map joining process, the state of the local submap is first transferred into the global coordinate frame and added into the global map state, common features presented in both the local and global maps are identified and then an estimator such as EKF or non-linear least squares is used to enforce the constraints.

Note that a key advantage of submap based approach is that it significantly reduces

the frequency of global map update. In addition to reducing the computational cost, submap based techniques may also improve the consistency of the globally estimated map [5] compared with EKF based approaches.

2.7 Curve Feature SLAM

While a lot of researchers in the field of feature based SLAM focus their research on reducing the computational cost, a group of researchers aim to improve the representation of the environment.

Although using point features is adequate in some scenarios, there are situations when relying solely on this representation becomes less appropriate. When the environment does not have sufficient structure to robustly extract point features e.g. in an underground mine environment [51], the information provided may not be enough for point feature SLAM. Furthermore, when point feature is used, only a small fraction of information available from popular sensors, such as laser range finders, is exploited. Much of the data that do not correspond to the expected features are discarded. Because of this, a number of algorithms which use more complex geometric primitives to represent the environment have emerged. We discuss some of the popular feature representation methods in this section. All these method assumes the segmentation process has already been completed.

2.7.1 Line feature SLAM

When line features are used in SLAM problem, one popular choice is to represent the line feature using the following form [45] [25]

$$\rho - x\cos(\alpha) - y\sin(\alpha) = 0 \tag{2.43}$$

where ρ is its distance from the origin and $\alpha \in (-\pi, \pi]$ is the direction of the normal passing through the origin. Suppose at robot pose V_{r_i} the robot observes n points $((x_1, y_1), \ldots, (x_n, y_n))$ that corresponding to a line feature V_f, the parameters of the line

$V_f = [\alpha,\ \rho]^T$ can be derived by minimizing the following cost function

$$\sum_{i=1}^{n}(\rho - x_i\cos(\alpha) - y_i\sin(\alpha))^2 \tag{2.44}$$

The above cost function can be minimized using the following parameters [25]

$$\begin{aligned}\alpha^* &= \frac{1}{2}\arctan\frac{-2S_{xy}}{S_{y^2} - S_{x^2}} \\ \rho^* &= \bar{x}\cos(\alpha^*) + \bar{y}\sin(\alpha^*)\end{aligned} \tag{2.45}$$

where

$$\begin{aligned}\bar{x} &= \frac{1}{n}\sum_{k=1}^{n}x_k, & \bar{y} &= \frac{1}{n}\sum_{k=1}^{n}y_k \\ S_{x^2} &= \sum_{k=1}^{n}(x_k - \bar{x})^2, & S_{y^2} &= \sum_{k=1}^{n}(y_k - \bar{y})^2 \\ S_{xy} &= \sum_{k=1}^{n}(x_i - \bar{x})(y_i - \bar{y}).\end{aligned} \tag{2.46}$$

The Jacobian matrix of V_f with respect to point $[x_k, y_k]^T$ is

$$C_f = \begin{bmatrix} C_f(1,1) & C_f(1,2) \\ C_f(2,1) & C_f(2,2) \end{bmatrix} \tag{2.47}$$

where

$$\begin{aligned}C_f(1,1) &= \frac{\cos(\alpha^*)}{n} - \bar{x}\sin(\alpha^*)C_f(2,1) + \bar{y}\cos(\alpha^*)C_f(2,1) \\ C_f(1,2) &= \frac{\sin(\alpha^*)}{n} - \bar{x}\sin(\alpha^*)C_f(2,2) + \bar{y}\cos(\alpha^*)C_f(2,2) \\ C_f(2,1) &= \frac{(\bar{y} - y_k)(S_{y^2} - S_{x^2} + 2S_{xy}(\bar{x} - x_k))}{(S_{y^2} - S_{x^2}) + 4S_{xy}^2} \\ C_f(2,2) &= \frac{(\bar{x} - x_k)(S_{y^2} - S_{x^2} - 2S_{xy}(\bar{y} - y_k))}{(S_{y^2} - S_{x^2}) + 4S_{xy}^2}\end{aligned} \tag{2.48}$$

Using the line parameters described above, line feature based SLAM problem can be

formulated.

2.7.2 SP-model

In SP model based SLAM [14], each geometric element has been associated with a reference coordinate system. Suppose an uncertain geometric feature is E, the location of the feature in SP-model based SLAM is described by the location vector $x_{WE} = [x, y, \phi]^T$ which is the location of the feature coordinate system with respect to the global coordinate system W.

The estimation of the location of a geometric element is denoted by $\hat{x}_{WE}$, and the estimation error is represented locally by the differential location vector d_E relative to the reference attached to the element. Thus, the true location of the element is

$$x_{WE} = \hat{x}_{WE} \oplus d_E \tag{2.49}$$

where $\otimes$ represents the composition of rotation and translation. Because a geometric element may contain symmetries, a null value is assigned to the corresponding degrees of freedom in d_E, because they do not represent an effective location error. From the effective elements in d_E, a *perturbation vector* p_E is formed which represents the effective location error. The relation between the p_E and d_E can be expressed by

$$d_E = B_E^T p_E \qquad p_E = B_E d_E \tag{2.50}$$

where B_E is a row selection matrix.

Using the parameters described above, SP-model based SLAM problem can be formulated.

2.7.3 B-Spline SLAM

In [64], Pedraza *et al.* developed the BS-SLAM where B-Splines are used to represent the environment.

A B-Spline curve of order k, is defined as

$$s(t) = \sum_{i=0}^{n} x_i \beta_{i,k}(t) \tag{2.51}$$

where x_i $(i = 0, ..., n)$ are the 2D control points, $\beta_{i,k}(t)$ are the normalized B-Spline basis functions of order k defined over the knot vector $T = [\varepsilon_0, ..., \varepsilon_{n+k}]$. A common form for clamped knot vector [65] for an order k spline is:

$$T = [\underbrace{0, ..., 0}_{k}, \varepsilon_k, ..., \varepsilon_n, \underbrace{1, ..., 1}_{k}] \tag{2.52}$$

with $0 \leq \varepsilon_k \leq ... \leq \varepsilon_n \leq 1$.

The basis functions $\beta_{i,k}(t)$ are governed by the Cox-de Boor recursion formulas [65].

$$\beta_{i,k}(t) = \begin{cases} 1, & \text{if } \varepsilon_i \leq t \leq \varepsilon_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{2.53}$$

and for $k > 1$

$$\beta_{i,k}(t) = \frac{t - \varepsilon_i}{\varepsilon_{i+k-1} - \varepsilon_i} \beta_{i,k-1}(t) + \frac{\varepsilon_{i+k} - t}{\varepsilon_{i+k} - \varepsilon_{i+1}} \beta_{i+1,k-1}(t) \tag{2.54}$$

where t is the “time" parameter representing which point of the spline is corresponding to. An example of a clamped spline with order 4 is shown in Fig. 2.3.

B-splines exhibit many interesting mathematical and geometrical properties. Some of them are enumerated here.

Affine invariance property

Any affine transformation of a B-Spline can be achieved simply by applying the transformation to its control points. This results in efficient transformation of a B-Spline. This property makes the transformation of B-Splines SLAM problem into point feature SLAM problem possible.

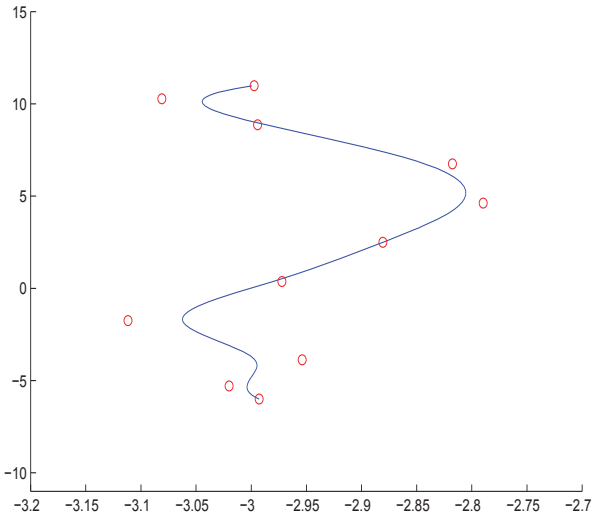


Figure 2.3: An example of B-Spline with order 4. The knot vector for this spline is [0,0,0,0,0.125,0.25,...0.75,0.875,1,1,1,1]. The control points are shown in red circle.

Approximating property

Another appeal of B-Spline curves is their ability for approximating noise-contaminated data. Using approximation techniques, the data points can be represented in the state vector.

One of the most common approximating technique is the parameterized spline fitting. In the parametrization process, for each data point

$$d_j = [p_j, q_j]^T,\ j = [1, ..., m]$$

a time parameter value t_j is assigned, for the whole set of data points a time parameter sequence $r_v = [t_1, ..., t_m]$ is formed. With a fixed time parameter sequence, the spline fitting problem becomes a minimization problem:

$$\min_V \sum_{j=0}^{m} \parallel \sum_{i=0}^{n} x_i \beta_{i,k}(t_j) - d_j \parallel^2 \tag{2.55}$$

where $V = [x_0, ..., x_n]^T$ and $x_i (i = 0, ..., n)$ are the control points. Define the positions of the raw data point as a matrix:

$$M = \begin{bmatrix} p_1 & \cdots & p_m \\ q_1 & \cdots & q_m \end{bmatrix}^T. \tag{2.56}$$

The least squares solution for (2.55) is:

$$V = [B^T B]^{-1} B^T M = \Phi M \tag{2.57}$$

where

$$\Phi = [B^T B]^{-1} B^T \tag{2.58}$$

and B is the *collocation matrix*:

$$B = \begin{bmatrix} \beta_{0,k}(t_1) & \cdots & \beta_{n,k}(t_1) \\ \vdots & \ddots & \vdots \\ \beta_{0,k}(t_m) & \cdots & \beta_{n,k}(t_m) \end{bmatrix}. \tag{2.59}$$

Differentiable property

Another important property of B-Spline is that B-Splines are differentiable. The derivative of a order k basis function $\beta_{i,k}(t)$ can be derived as

$$\frac{\partial \beta_{i,k}(t)}{\partial t} = \frac{k-1}{\varepsilon_{i+k-1} - \varepsilon_i} \beta_{i,k-1}(t) - \frac{k-1}{\varepsilon_{i+k} - \varepsilon_{i+1}} \beta_{i+1,k-1}(t). \tag{2.60}$$

Hence the derivative of a B-spline of order k is

$$\frac{ds(t)}{dt} = (k-1) \sum_{i=0}^{n} \frac{x_i - x_{i-1}}{\xi_{i+k-1} - \xi_i} \beta_{i,k-1}(t). \tag{2.61}$$

In [64], B-Splines are represented by control points in the state vector and EKF is applied to solve the B-Spline based SLAM problem.

2.8 Summary

This chapter presents a brief review of some of the existing feature based SLAM algorithm. It begins with introducing the point feature SLAM optimization formulation. Typically, this optimization is solved by the least squares technique. In particular, two popular least squares estimation techniques (Gauss-Newton and Levenberg-Marquardt) are introduced. Algorithms (SAM and SBA) which reduces the computational cost of these two least square techniques are also reviewed. Some other complex feature representation methods (line features, SP model and B-Splines) are also briefly covered in this chapter.

3 A Semi-definite Relaxation Based Approach for Point Feature SLAM Problems

3.1 Introduction

This chapter presents a convex optimization based approach for solving point feature based simultaneous localization and mapping (SLAM) problems. It is demonstrated that, when covariance matrix is spherical, using a new representation for robot orientation, the point feature SLAM problem can be formulated as a quadratically constrained quadratic programming (QCQP) problem. The QCQP problem can then be relaxed to obtain a semi-definite programming (SDP) problem. As SDP is a convex optimization problem, a unique solution can be obtained without initial state estimates. This solution is then used to construct a candidate solution to the SLAM problem.

Although the new formulation is an approximation to the least squares SLAM formulation, it has been proved that, when all the measurements contain no noise, the semi-definite relaxation (SDR) result is the optimal solution to the point feature SLAM least squares formulation. It will also be shown in the next chapter using computer simulations and experimental data that, the candidate solution obtained is very close to the true SLAM solution obtained through nonlinear least squares.

The chapter is organized as follows. Section 3.2 provides an overview of the proposed approach. Section 3.3 proposes a new state representation which simplifies the measurement equation. Section 3.4 analyzes why the objective function does not contain product of the state variables when the covariance matrix is spherical and proposes a covariance matrix approximation method. Section 3.5 describe how the transformation of SLAM

problem to QCQP is achieved. Sections 3.6 and 3.7 briefly review the SDR technique, illustrate how the QCQP is converted to a SDP and discuss how to extract a candidate solution to original SLAM problem from the SDP result. Finally, Section 3.8 summarizes the chapter.

3.2 Overview of the Proposed Approach

This chapter proposes the SDR based approach to solve the SLAM problems. The major steps in the proposed approach are given in Algorithm 3.1. The details of each step are provided in the following sections.

Algorithm 3.1: SDR based approach for point feature based SLAM

Given a data association already completed dataset.

Step 1: Introducing new state variables (Section 3.3).
Step 2: Approximating the covariance matrices to be spherical and reformulate the SLAM problem (Section 3.4).
Step 3: Transform the proposed SLAM formulations into QCQP (Section 3.5).
Step 4: Convert the QCQP problem into SDP using the SDR technique and solve the SDP using convex optimization techniques (Section 3.6).
Step 5: From SDP result, construct a candidate solution to the SLAM problem. Refine the candidate solution to obtain the optimal solution if necessary (Section 3.7).

3.3 New State Variables: $\cos(\phi_{r_i})$ and $\sin(\phi_{r_i})$

In the least squares SLAM formulation, the orientation component of the robot pose r_i is represented by ϕ_{r_i}. One limitation of this representation is that, the observation function (2.3) and function of odometry (2.8) involve $\cos(\phi_{r_i})$ and $\sin(\phi_{r_i})$. This makes the observation and odometry functions nonlinear.

Here, it is proposed to represent the robot orientation using two variables $c_{r_i} = \cos(\phi_{r_i})$ and $s_{r_i} = \sin(\phi_{r_i})$. Hence the robot pose is represented by

$$\tilde{X}_{r_i} = \left[\begin{array}{c} x_{r_i}, y_{r_i}, c_{r_i}, s_{r_i} \end{array} \right]^T$$

and the new state representation is

$$\tilde{V} = [\tilde{X}_{r_1}^T, \ldots, \tilde{X}_{r_p}^T, X_{f_1}^T, \ldots, X_{f_N}^T]^T.$$

Using the new state representation, unlike (2.3) and (2.8), the functions of observation and odometry can be written as

$$\tilde{H}^{z_j^i}(\tilde{V}) = \left[\begin{array}{c} c_{r_i}(x_{f_j} - x_{r_i}) + s_{r_i}(y_{f_j} - y_{r_i}) \\ -s_{r_i}(x_{f_j} - x_{r_i}) + c_{r_i}(y_{f_j} - y_{r_i}) \end{array} \right] \tag{3.1}$$

$$\tilde{H}^{\tilde{o}_i^{x,y}}(\tilde{V}) = \left[\begin{array}{c} c_{r_i}(x_{r_{i+1}} - x_{r_i}) + s_{r_i}(y_{r_{i+1}} - y_{r_i}) \\ -s_{r_i}(x_{r_{i+1}} - x_{r_i}) + c_{r_i}(y_{r_{i+1}} - y_{r_i}) \end{array} \right] \tag{3.2}$$

$$\tilde{H}^{\tilde{o}_i^{\phi}}(\tilde{V}) = \left[\begin{array}{c} c_{r_i}c_{r_{i+1}} + s_{r_i}s_{r_{i+1}} \\ -s_{r_i}c_{r_{i+1}} + c_{r_i}s_{r_{i+1}} \end{array} \right] \tag{3.3}$$

$$\tilde{H}^{\tilde{o}_i}(\tilde{V}) = \left[\begin{array}{c} \tilde{H}^{\tilde{o}_i^{x,y}}(\tilde{V}) \\ \tilde{H}^{\tilde{o}_i^{\phi}}(\tilde{V}) \end{array} \right]. \tag{3.4}$$

Clearly, (3.1)-(3.4) are bilinear functions of the state variables. Of course, the following quadratic constraints are needed:

$$c_{r_i}^2 + s_{r_i}^2 = 1, \ i = 1, \ldots, p. \tag{3.5}$$

At this point, the new state variable representation does not simplify the SLAM problem

much as (3.1)-(3.4) still contains product of the state variables which makes the objective function biquadratic and the optimization problem hard to solve. However, when applying some approximations, the objective function will become quadratic as shown below in Section 3.4.

3.4 Approximating the Covariance Matrices to be Spherical

In [33] and [67] it has been observed that, when covariance matrices are spherical the objective function does not contain product of the state variables. Hence the objective function becomes quadratic. Define

$$\delta = \begin{bmatrix} x_{f_j} - x_{r_i} \\ y_{f_j} - y_{r_i} \end{bmatrix} \tag{3.6}$$

$$\triangle = \begin{bmatrix} x_{r_{i+1}} - x_{r_i} \\ y_{r_{i+1}} - y_{r_i} \end{bmatrix} \tag{3.7}$$

$$\Theta = \begin{bmatrix} c_{r_{i+1}} \\ s_{r_{i+1}} \end{bmatrix} \tag{3.8}$$

$$R = \begin{bmatrix} c_{r_i} & -s_{r_i} \\ s_{r_i} & c_{r_i} \end{bmatrix}, \tag{3.9}$$

then (3.1), (3.2) and (3.3) can be written as

$$\tilde{H}^{z_j^i}(\tilde{V}) = R^T \delta \tag{3.10}$$

$$\tilde{H}^{\tilde{o}_i^{x,y}}(\tilde{V}) = R^T \triangle \tag{3.11}$$

$$\tilde{H}^{\tilde{o}_i^{\phi}}(\tilde{V}) = R^T \Theta. \tag{3.12}$$

Consider a single feature observation z_j^i with spherical covariance matrix

$$\tilde{P}_{z_j^i} = diag[\lambda_{z_j^i}, \lambda_{z_j^i}] = \lambda_{z_j^i} I,\ \lambda_{z_j^i} > 0 \tag{3.13}$$

one term of the objective function in (2.11) related to z_j^i is

$$\begin{aligned} & (z_j^i - \tilde{H}^{z_j^i}(\tilde{V}))^T \tilde{P}_{z_j^i}^{-1} (z_j^i - \tilde{H}^{z_j^i}(\tilde{V})) \\ = \; & (z_j^i - R^T\delta)^T \tilde{P}_{z_j^i}^{-1} (z_j^i - R^T\delta) \\ = \; & [R^T(Rz_j^i - \delta)]^T \tilde{P}_{z_j^i}^{-1} [R^T(Rz_j^i - \delta)] \\ = \; & (Rz_j^i - \delta)^T R \tilde{P}_{z_j^i}^{-1} R^T (Rz_j^i - \delta). \end{aligned} \tag{3.14}$$

From (3.13) we can have

$$R\tilde{P}_{z_j^i}^{-1}R^T = \tilde{P}_{z_j^i}^{-1} = \frac{1}{\lambda_{z_j^i}} I \tag{3.15}$$

and (3.14) now becomes

$$\frac{1}{\lambda_{z_j^i}} (Rz_j^i - \delta)^T (Rz_j^i - \delta). \tag{3.16}$$

Since z_j^i is the actual observation, $(Rz_j^i - \delta)$ is linear in terms of the variables $x_{f_j}, y_{f_j}, x_{r_i}, y_{r_i}, c_{r_i}, s_{r_i}$, thus (3.16) is a quadratic function.

Similarly, when odometry is available, consider the relative robot position measurement with spherical covariance matrix

$$\tilde{P}_{\tilde{o}_i^{x,y}} = diag[\lambda_{\tilde{o}_i^{x,y}}, \lambda_{\tilde{o}_i^{x,y}}] = \lambda_{\tilde{o}_i^{x,y}} I, \;\; \lambda_{\tilde{o}_i^{x,y}} > 0. \tag{3.17}$$

One term of the objective function in (2.12) related to the translation part of the odometry is

$$\begin{aligned} & (\tilde{o}_i^{x,y} - \tilde{H}^{\tilde{o}_i^{x,y}}(\tilde{V}))^T \tilde{P}_{\tilde{o}_i^{x,y}}^{-1} (\tilde{o}_i^{x,y} - \tilde{H}^{\tilde{o}_i^{x,y}}(\tilde{V})) \\ = \; & (\tilde{o}_i^{x,y} - R^T\triangle)^T \tilde{P}_{\tilde{o}_i^{x,y}}^{-1} (\tilde{o}_i^{x,y} - R^T\triangle) \\ = \; & \frac{1}{\lambda_{\tilde{o}_i^{x,y}}} (R\tilde{o}_i^{x,y} - \triangle)^T (R\tilde{o}_i^{x,y} - \triangle) \end{aligned} \tag{3.18}$$

which is also a quadratic function. The true covariance matrices $P_{z_j^i}$ and P_{o_i} may not be spherical. We propose to use the following to approximate these matrices.

$$\begin{aligned} \lambda_{z_j^i} &= max(P_{z_j^i}(1,1), \; P_{z_j^i}(2,2)), \\ \lambda_{\tilde{o}_i^{x,y}} &= max(P_{o_i}(1,1), \; P_{o_i}(2,2)). \end{aligned} \tag{3.19}$$

3.5 Transform the Proposed SLAM Formulation into QCQP

3.5.1 When odometry is unavailable

The proposed SLAM problem for this case is

$$\begin{aligned} &\min \sum_{i=0}^{p} \sum_{j=1}^{k_i} (z_j^i - \tilde{H}^{z_j^i}(\tilde{V}))^T \tilde{P}_{z_j^i}^{-1} (z_j^i - \tilde{H}^{z_j^i}(\tilde{V})) \\ &s.t. \quad c_{r_i}^2 + s_{r_i}^2 = 1, \ i = 1, \ldots, p \end{aligned} \tag{3.20}$$

From (3.1) and (3.16), the minimization problem (3.20) can be rewritten as

$$\begin{aligned} \min \sum_{i=0}^{p} \sum_{j=1}^{k_i} \frac{1}{\lambda_{z_j^i}} [&(z_{x_j^i} c_{r_i} - z_{y_j^i} s_{r_i} - x_{f_j} + x_{r_i})^2 \\ &+ (z_{x_j^i} s_{r_i} + z_{y_j^i} c_{r_i} - y_{f_j} + y_{r_i})^2] \\ s.t. \ \ c_{r_i}^2 + s_{r_i}^2 &= 1, i = 1, \ldots, p \end{aligned} \tag{3.21}$$

In (3.21), when $i > 0$, $(z_{x_j^i} c_{r_i} - z_{y_j^i} s_{r_i} - x_{f_j} + x_{r_i})^2 + (z_{x_j^i} s_{r_i} + z_{y_j^i} c_{r_i} - y_{f_j} + y_{r_i})^2$ can be rewritten in quadratic form in terms of state vector $\tilde{V}$

$$\tilde{V}^T T_{z_i, f_j}^T Q_{j,i} T_{z_i, f_j} \tilde{V} \tag{3.22}$$

where

$$Q_{j,i} = \begin{bmatrix} 1 & 0 & z_{x_j^i} & -z_{y_j^i} & -1 & 0 \\ 0 & 1 & z_{y_j^i} & z_{x_j^i} & 0 & -1 \\ z_{x_j^i} & z_{y_j^i} & z_{x_j^i}^2 + z_{y_j^i}^2 & 0 & -z_{x_j^i} & -z_{y_j^i} \\ -z_{y_j^i} & z_{x_j^i} & 0 & z_{x_j^i}^2 + z_{y_j^i}^2 & z_{y_j^i} & -z_{x_j^i} \\ -1 & 0 & -z_{x_j^i} & z_{y_j^i} & 1 & 0 \\ 0 & -1 & -z_{y_j^i} & -z_{x_j^i} & 0 & 1 \end{bmatrix} \tag{3.23}$$

T_{z_i,f_j} is a selection matrix to select i-th robot pose and j-th feature from the state vector $\tilde{V}$, i.e.,

$$T_{z_i,f_j}\tilde{V} = [x_{r_i}, y_{r_i}, c_{r_i}, s_{r_i}, x_{f_j}, y_{f_j}]^T.$$

In (3.21), when $i = 0$, we have $x_{r_i} = 0, y_{r_i} = 0$, $c_{r_i} = 1, s_{r_i} = 0$, thus

$$\sum_{j=1}^{k_i} \frac{1}{\lambda_{z_j^i}} [(z_{x_j^i} c_{r_i} - z_{y_j^i} s_{r_i} - x_{f_j} + x_{r_i})^2 + (z_{x_j^i} s_{r_i} + z_{y_j^i} c_{r_i} - y_{f_j} + y_{r_i})^2]$$

becomes

$$\sum_{j=1}^{k_0} \frac{1}{\lambda_{z_j^0}} [(z_{x_j^0} - x_{f_j})^2 + (z_{y_j^0} - y_{f_j})^2] \tag{3.24}$$

which is

$$\sum_{j=1}^{k_0} \frac{1}{\lambda_{z_j^0}} (\tilde{V}^T T_{z_0,f_j}^T Q_{j,0} T_{z_0,f_j} \tilde{V} + a_{j,0}^T T_{z_0,f_j} \tilde{V} + b_{j,0}) \tag{3.25}$$

where $Q_{j,0} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $a_{j,0} = \begin{bmatrix} -2z_{x_j^0} \\ -2z_{y_j^0} \end{bmatrix}$, $T_{z_0,f_j}\tilde{V} = \begin{bmatrix} x_{f_j} \\ y_{f_j} \end{bmatrix}$ and $b_{j,0} = (z_{x_j^0})^2 + (z_{y_j^0})^2$. It can be observed that, in (3.25), the linear terms are related to the observations made from the first pose, while the quadratic terms are related to the observations from the other poses. Similarly, the constraints in (3.21) can also be written as

$$\tilde{V}^T T_{c_i,s_i}^T T_{c_i,s_i} \tilde{V} = 1, \quad i = 1, \ldots, p \tag{3.26}$$

where T_{c_i,s_i} is a selection matrix such that c_{r_i}, s_{r_i} are selected from $\tilde{V}$, i.e.,

$$T_{c_i,s_i}\tilde{V} = \begin{bmatrix} c_{r_i} & s_{r_i} \end{bmatrix}^T \tag{3.27}$$

Define

$$\begin{aligned}
\tilde{Q}_0 &= \sum_{i=0}^{p}\sum_{j=1}^{k_i}\frac{1}{\lambda_{z_j^i}}T_{z_i,f_j}^T Q_{j,i}T_{z_i,f_j}\\
\tilde{a}_0^T &= \sum_{j=1}^{k_0}\frac{1}{\lambda_{z_j^0}}a_{j,0}^T T_{z_0,f_j}\\
\tilde{b}_0 &= \sum_{j=1}^{k_0}\frac{1}{\lambda_{z_j^0}}b_{j,0}\\
\tilde{Q}_i &= T_{c_i,s_i}^T T_{c_i,s_i}
\end{aligned}$$

Then (3.21) becomes

$$\begin{aligned}
\min\quad & \tilde{V}^T\tilde{Q}_0\tilde{V} + \tilde{a}_0^T\tilde{V} + \tilde{b}_0\\
s.t.\quad & \tilde{V}^T\tilde{Q}_i\tilde{V} = 1,\ i = 1,\ldots,p
\end{aligned} \tag{3.28}$$

which is in standard QCQP form [2]. It can be notices that $\tilde{Q}_0$ is only related to the robot to feature measurement and the translation part of the relative pose measurement. The $\tilde{Q}_i$ part is only related to the orientation part of the relative pose measurement. Note that $\tilde{Q}_0$ and $\tilde{Q}_i$ are all sparse matrices.

3.5.2 When odometry is available

Unlike the relative translation part, the orientation part of the relative pose constraint can not be approximated directly. This is because when deriving the covariance matrix for new relative robot orientation $\tilde{o}_i^\phi$ using the formula

$$\tilde{C}_{\tilde{o}_i^\phi} = \frac{\partial\tilde{o}_i^\phi}{\partial o_{i,j}^\phi}C_{o_i^\phi}(\frac{\partial\tilde{o}_i^\phi}{\partial o_i^\phi})^T \tag{3.29}$$

where $C_{o_i^\phi}$ is the uncertainty for the original relative robot orientation o_i^ϕ. The covariance matrix $\tilde{C}_{\tilde{o}_{i,j}^\phi}$ becomes singular.

Therefore, we therefore propose to treat the relative robot orientation as constraints. Suppose the relative robot orientation is within 99% confidence interval, the bound for

relative robot orientation can be written as $[o_i^{\phi}-3\sigma_{\phi_i}, o_i^{\phi}+3\sigma_{\phi_i}]$ where σ_{ϕ_i} is approximated by:

$$\sigma_{\phi_i} = \sqrt{P_{o_i}(3,3)} \tag{3.30}$$

The bound for c_{r_i} and s_{r_i} can be then derived as

$$\begin{aligned} o_{\phi_i}^{cu} &\geq c_{r_{i+1}}c_{r_i} + s_{r_{i+1}}s_{r_i} \geq o_{\phi_i}^{cl} \\ o_{\phi_i}^{su} &\geq s_{r_{i+1}}c_{r_i} - c_{r_{i+1}}s_{r_i} \geq o_{\phi_i}^{sl} \end{aligned} \tag{3.31}$$

where

$$\begin{aligned} o_{\phi_i}^{cu} &= \max\{\cos(\phi) : \phi \in [o_{\phi_i} - 3\sigma_{\phi_i}, o_{\phi_i} + 3\sigma_{\phi_i}]\} \\ o_{\phi_i}^{cl} &= \min\{\cos(\phi) : \phi \in [o_{\phi_i} - 3\sigma_{\phi_i}, o_{\phi_i} + 3\sigma_{\phi_i}]\} \\ o_{\phi_i}^{su} &= \max\{\sin(\phi) : \phi \in [o_{\phi_i} - 3\sigma_{\phi_i}, o_{\phi_i} + 3\sigma_{\phi_i}]\} \\ o_{\phi_i}^{sl} &= \min\{\sin(\phi) : \phi \in [o_{\phi_i} - 3\sigma_{\phi_i}, o_{\phi_i} + 3\sigma_{\phi_i}]\} \end{aligned} \tag{3.32}$$

Here we restrict the error in orientation to be within 3-sigma bound, so we choose the lower/upper bound of c_{r_i} and s_{r_i} this way.

After the reformulation and approximation explained in Section 3.4, the proposed SLAM problem for odometry available case can be written as:

$$\begin{aligned} \min \; & \sum_{i=0}^{p}\sum_{j=1}^{k_i}(z_j^i - \tilde{H}^{z_j^i}(\tilde{V}))^T \tilde{P}_{z_j^i}^{-1}(z_j^i - \tilde{H}^{z_j^i}(\tilde{V})) \\ & + \sum_{i=0}^{p-1}(\tilde{o}_i^{x,y} - \tilde{H}^{\tilde{o}_i^{x,y}}(\tilde{V}))^T \tilde{P}_{\tilde{o}_i^{x,y}}^{-1}(\tilde{o}_i^{x,y} - \tilde{H}^{\tilde{o}_i^{x,y}}(\tilde{V})) \\ s.t. \quad & (c_{r_i})^2 + (s_{r_i})^2 = 1, \; i = 1, \ldots, p \\ & o_{\phi_i}^{cu} \geq c_{r_{i+1}}c_{r_i} + s_{r_{i+1}}s_{r_i} \geq o_{\phi_i}^{cl}, \; i = 0, \ldots, p-1 \\ & o_{\phi_i}^{su} \geq s_{r_{i+1}}c_{r_i} - c_{r_{i+1}}s_{r_i} \geq o_{\phi_i}^{sl}, \; i = 0, \ldots, p-1 \end{aligned} \tag{3.33}$$

Similar to the odometry unavailable case, the minimization problem can be formulated

as

$$\begin{aligned}
\min & \sum_{i=0}^{p}\sum_{j=1}^{k_i}\frac{1}{\lambda_{z_j^i}}[(z_{x_j^i}c_{r_i}-z_{y_j^i}s_{r_i}-x_{f_j}+x_{r_i})^2 \\
& +(z_{x_j^i}s_{r_i}+z_{y_j^i}c_{r_i}-y_{f_j}+y_{r_i})^2] \\
& +\sum_{i=0}^{p-1}\frac{1}{\lambda_{\tilde{o}_i^{x,y}}}[(o_{x_i}c_{r_i}-o_{y_i}s_{r_i}-x_{r_{i+1}}+x_{r_i})^2 \\
& +(o_{x_i}s_{r_i}+o_{y_i}c_{r_i}-y_{r_{i+1}}+y_{r_i})^2] \\
s.t. \ & (c_{r_i})^2+(s_{r_i})^2=1,\ i=1,\dots,p \\
& o_{\phi_i}^{cu}\geq c_{r_{i+1}}c_{r_i}+s_{r_{i+1}}s_{r_i}\geq o_{\phi_i}^{cl},\ i=0,\dots,p-1 \\
& o_{\phi_i}^{su}\geq s_{r_{i+1}}c_{r_i}-c_{r_i}s_{r_{i-1}}\geq o_{\phi_i}^{sl},\ i=0,\dots,p-1
\end{aligned} \tag{3.34}$$

and (3.34) can be converted into the following problem

$$\begin{aligned}
\min & \sum_{i=0}^{p}\sum_{j=1}^{k_i}\frac{1}{\lambda_{z_j^i}}\tilde{V}^TT_{z_i,f_j}^TQ_{j,i}T_{z_i,f_j}\tilde{V}+\sum_{i=0}^{p-1}\frac{1}{\lambda_{\tilde{o}_i^{x,y}}}\tilde{V}^TT_{o_i}^TQ_{o_i}T_{o_i}\tilde{V} \\
& +\sum_{j=1}^{k_0}\frac{1}{\lambda_{z_j^0}}(a_{j,0}^TT_{z_0,f_j}\tilde{V}+b_{j,0})+\frac{1}{\lambda_{\tilde{o}_0^{x,y}}}((a_0^1)^TT_{o_0}\tilde{V}+b_0^1) \\
s.t. \ & \tilde{V}^TT_{c_i,s_i}^TT_{c_i,s_i}\tilde{V}=1,\ i=1,\dots,p \\
& o_{\phi_i}^{cu}\geq\tilde{V}^TT_{c_{i+1}}^TT_{c_i}\tilde{V}+\tilde{V}^TT_{s_{i+1}}^TT_{s_i}\tilde{V}\geq o_{\phi_i}^{cl}, \\
& i=0,\dots,p-1 \\
& o_{\phi_i}^{su}\geq\tilde{V}^TT_{s_{i+1}}^TT_{c_i}\tilde{V}-\tilde{V}^TT_{c_{i+1}}^TT_{s_i}\tilde{V}\geq o_{\phi_i}^{sl}, \\
& i=0,\dots,p-1
\end{aligned} \tag{3.35}$$

where for $i\neq 0$, $Q_{j,i}$ is the same as defined in (3.23) and

$$Q_{o_i}=\begin{bmatrix}
1 & 0 & o_{x_i} & -o_{y_i} & -1 & 0 \\
0 & 1 & o_{y_i} & o_{x_i} & 0 & -1 \\
o_{x_i} & o_{y_i} & o_{x_i}^2+o_{y_i}^2 & 0 & -o_{x_i} & -o_{y_i} \\
-o_{y_i} & o_{x_i} & 0 & o_{x_i}^2+o_{y_i}^2 & o_{y_i} & -o_{x_i} \\
-1 & 0 & -o_{x_i} & o_{y_i} & 1 & 0 \\
0 & -1 & -o_{y_i} & -o_{x_i} & 0 & 1
\end{bmatrix}, \tag{3.36}$$

T_{o_i}, T_{z_i,f_j}, T_{c_i,s_i}, $T_{c_{i+1}}$, T_{c_i}, $T_{s_{i+1}}$, T_{s_i} are selection matrices of appropriate dimensions such that

$$\begin{aligned}
T_{o_i}\tilde{V} &= \begin{bmatrix} x_{r_i} & y_{r_i} & c_{r_i} & s_{r_i} & x_{r_{i+1}} & y_{r_{i+1}} \end{bmatrix}^T \\
T_{z_i,f_j}\tilde{V} &= \begin{bmatrix} x_{r_i} & y_{r_i} & c_{r_i} & s_{r_i} & x_{f_j} & y_{f_j} \end{bmatrix}^T \\
T_{c_i,s_i}\tilde{V} &= \begin{bmatrix} c_{r_i} & s_{r_i} \end{bmatrix}^T \\
T_{c_{i+1}}\tilde{V} &= c_{r_{i+1}} \\
T_{c_i}\tilde{V} &= c_{r_i} \\
T_{s_{i+1}}\tilde{V} &= s_{r_{i+1}} \\
T_{s_i}\tilde{V} &= s_{r_i}.
\end{aligned} \tag{3.37}$$

When $i = 0$,

$$\begin{aligned}
Q_{j,0} = Q_{o_0} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \\
T_{z_0,f_j}\tilde{V} &= \begin{bmatrix} x_{f_j} & y_{f_j} \end{bmatrix}^T, \\
T_{o_0}\tilde{V} &= \begin{bmatrix} x_{r_1} & y_{r_1} \end{bmatrix}^T, \\
a_{j,0} &= \begin{bmatrix} -2z_{x_j^0} & -2z_{y_j^0} \end{bmatrix}^T, \\
b_{j,0} &= (z_{x_j^0})^2 + (z_{y_j^0})^2, \\
a_0^1 &= \begin{bmatrix} -2o_{x_1^0} & -2o_{y_1^0} \end{bmatrix}^T, \\
b_0^1 &= (o_{x_1^0})^2 + (o_{y_1^0})^2.
\end{aligned} \tag{3.38}$$

Define

$$\begin{aligned}
\hat{Q}_0 &= \sum_{i=0}^{p}\sum_{j=1}^{k_i}\frac{1}{\lambda_{z_j^i}}T_{z_i,f_j}^T Q_{j,i} T_{z_i,f_j} + \sum_{i=0}^{p-1}\frac{1}{\lambda_{\tilde{o}_i^{x,y}}}(T_{o_i}^T Q_{o_i} T_{o_i}) \\
\tilde{Q}_i &= T_{c_i,s_i}^T T_{c_i,s_i} \\
\hat{Q}_{ci} &= T_{c_{i+1}}^T T_{c_i} + T_{s_{i+1}}^T T_{s_i} \\
\hat{Q}_{si} &= T_{s_{i+1}}^T T_{c_i} - T_{c_{i+1}}^T T_{s_i} \\
\hat{a}_0^T &= \sum_{j=1}^{k_0}\frac{1}{\lambda_{z_j^0}}a_{j,0}^T T_{z_0,f_j} + \frac{1}{\lambda_{\tilde{o}_0^{x,y}}}(a_0^1)^T T_{o_0} \\
\hat{b}_0 &= \sum_{j=1}^{k_0}\frac{1}{\lambda_{z_j^0}}b_{j,0} + \frac{1}{\lambda_{\tilde{o}_0^{x,y}}}b_0^1
\end{aligned}$$

Then (3.33) becomes

$$\begin{aligned}
\min \quad & \tilde{V}^T\hat{Q}_0\tilde{V} + \hat{a}_0^T\tilde{V} + \hat{b}_0 \\
s.t. \quad & \tilde{V}^T\tilde{Q}_i\tilde{V} = 1, i = 1,\ldots,p \\
& o_{\phi_i}^{cu} \geq \tilde{V}^T\hat{Q}_{ci}\tilde{V} \geq o_{\phi_i}^{cl},\ i = 0,\ldots,p-1 \\
& o_{\phi_i}^{su} \geq \tilde{V}^T\hat{Q}_{si}\tilde{V} \geq o_{\phi_i}^{sl},\ i = 0,\ldots,p-1
\end{aligned} \tag{3.39}$$

which is in standard QCQP form. Again, in (3.39), $\hat{Q}_0$, $\tilde{Q}_i$, $\hat{Q}_{ci}$ and $\hat{Q}_{si}$ are sparse matrices.

3.6 Apply SDR to Transfer the QCQP into Convex Optimization Problem

From Section 3.5 the SLAM problems (3.20) and (3.33) have been transformed into standard QCQP (3.28) and (3.39). In this section, we show how to apply SDR technique [50] [2] on standard QCQP.

Consider a QCQP of the standard form [2]

$$\begin{aligned} \min \quad & \tilde{V}^T Q_0 \tilde{V} + a_0^T \tilde{V} + b_0 \\ s.t. \quad & \tilde{V}^T Q_i \tilde{V} + a_i^T \tilde{V} \leq b_i, i \in \mathcal{I} \\ & \tilde{V}^T Q_i \tilde{V} + a_i^T \tilde{V} = b_i, i \in \mathcal{E} \end{aligned} \tag{3.40}$$

where $\mathcal{I}$ is the set of indices of inequality constraints, $\mathcal{E}$ is the set of indices of equality constraints and $\mathcal{I} \cup \mathcal{E} = \{1, \ldots, m\}$. The matrices Q_i are assumed to be symmetric. Because

$$\tilde{V}^T Q \tilde{V} = Tr(\tilde{V}^T Q \tilde{V}) = Tr(Q \tilde{V} \tilde{V}^T), \tag{3.41}$$

problem (3.40) is equivalent to

$$\begin{aligned} \min \quad & Tr(Q_0 \tilde{V} \tilde{V}^T) + a_0^T \tilde{V} + b_0 \\ s.t. \quad & Tr(Q_i \tilde{V} \tilde{V}^T) + a_i^T \tilde{V} \leq b_i, i \in \mathcal{I} \\ & Tr(Q_i \tilde{V} \tilde{V}^T) + a_i^T \tilde{V} = b_i, i \in \mathcal{E} \end{aligned} \tag{3.42}$$

Define a new variable $X = \tilde{V} \tilde{V}^T$, (3.42) can be written as:

$$\begin{aligned} \min \quad & Tr(Q_0 X) + a_0^T \tilde{V} + b_0 \\ s.t. \quad & Tr(Q_i X) + a_i^T \tilde{V} \leq b_i, i \in \mathcal{I} \\ & Tr(Q_i X) + a_i^T \tilde{V} = b_i, i \in \mathcal{E} \\ & X - \tilde{V} \tilde{V}^T = 0. \end{aligned} \tag{3.43}$$

In (3.43), the only difficult constraint is the last constraint $X - \tilde{V} \tilde{V}^T = 0$, which is non-convex (the objective function and all other constraints in (3.43) are linear thus convex). Therefore SDR relaxes this problem into a convex problem by replacing the

last constraint with a (convex) positive semi-definite constraint $X - \tilde{V}\tilde{V}^T \succeq 0$:

$$\begin{aligned} \min \quad & Tr(Q_0 X) + a_0^T \tilde{V} + b_0 \\ s.t. \quad & Tr(Q_i X) + a_i^T \tilde{V} \leq b_i, i \in \mathcal{I} \\ & Tr(Q_i X) + a_i^T \tilde{V} = b_i, i \in \mathcal{E} \\ & X - \tilde{V}\tilde{V}^T \succeq 0 \end{aligned} \tag{3.44}$$

Since $X - \tilde{V}\tilde{V}^T \succeq 0$ is equivalent to $\tilde{X} = \begin{bmatrix} 1 & \tilde{V}^T \\ \tilde{V} & X \end{bmatrix} \succeq 0$, (3.44) may be alternatively written in the form

$$\begin{aligned} \min \quad & Tr(\check{Q}_0 \tilde{X}) \\ s.t. \quad & Tr(\check{Q}_i \tilde{X}) \leq 0, i \in \mathcal{I} \\ & Tr(\check{Q}_i \tilde{X}) = 0, i \in \mathcal{E} \\ & \tilde{X} \succeq 0 \end{aligned} \tag{3.45}$$

where $\check{Q}_i = \begin{bmatrix} -b_i & a_i^T/2 \\ a_i/2 & Q_i \end{bmatrix}$, $\check{Q}_0 = \begin{bmatrix} b_0 & a_0^T/2 \\ a_0/2 & Q_0 \end{bmatrix}$.

Unique solution $\tilde{X}^*$ to (3.45) can then be obtained using SDP packages such as SeDumi [75] and DSDP [9].

Remark 3.1. It should be noted that, when $\tilde{V}^*$ and X^* of the SDP solution $\tilde{X}^*$ satisfies $X^* - \tilde{V}^* \tilde{V}^{*^T} = 0$, the SDP solution is the optimal solution to the QCQP problem (3.40). However, $\tilde{V}^*$ and X^* from the SDP solution $\tilde{X}^*$ may not satisfy $X^* - \tilde{V}^* \tilde{V}^{*^T} = 0$. Whenever $X^* \neq \tilde{V}^* \tilde{V}^{*^T}$, $\tilde{V}^*$ may not be a feasible solution to the QCQP problem (3.40) as the constraints in (3.40) may not hold.

3.7 Obtain Feasible Solution

In previous sections, we have converted the SLAM problem into QCQP form and applied SDR technique to the QCQP problem such that unique solution $\tilde{X}^*$ to the relaxed convex optimization problem can be derived. As $\tilde{V}^*$ from the SDP result $\tilde{X}^*$ may not be a feasible solution to the QCQP problem, now we discuss how to get a candidate solution to SLAM

problem (2.11) or (2.12).

In general, deriving a feasible solution to QCQP from the SDP solution is non-trivial and problem dependant [50]. However, for the SLAM problem, the process of finding a candidate solution is relatively easy. In the proposed SLAM formulations, since the constraints (3.31) are introduced due to the orientation part of the odometry, the violation of these constraints only means that the solution contradicts the odometry information. Thus, the only constraint of concern are

$$c_{r_i}^2 + s_{r_i}^2 = 1,\ i = 1, \ldots, p. \tag{3.46}$$

Therefore, any state vector $\tilde{V}$ which satisfies (3.46) is a candidate solution. One simple and intuitive way to get a candidate solution from $\tilde{V}^*$ is to normalize the angular part of $\tilde{V}^*$. Suppose the SDP solution is:

$$\tilde{V}^* = [x_{r_1}^*, y_{r_1}^*, c_{r_1}^*, s_{r_1}^*, \ldots, x_{r_p}^*, y_{r_p}^*, c_{r_p}^*, s_{r_p}^*, x_{f_1}^*, y_{f_1}^*, \ldots, x_{f_N}^*, y_{f_N}^*]^T,$$

the angular part of the state vector can be normalized to get a candidate solution

$$\begin{aligned}\tilde{V}^f =& [x_{r_1}^*, y_{r_1}^*, \frac{c_{r_1}^*}{\sqrt{(c_{r_1}^*)^2 + (s_{r_1}^*)^2}}, \frac{s_{r_1}^*}{\sqrt{(c_{r_1}^*)^2 + (s_{r_1}^*)^2}}, \ldots, x_{r_p}^*, y_{r_p}^*, \\ & \frac{c_{r_p}^*}{\sqrt{(c_{r_p}^*)^2 + (s_{r_p}^*)^2}}, \frac{s_{r_p}^*}{\sqrt{(c_{r_p}^*)^2 + (s_{r_p}^*)^2}}, x_{f_1}^*, y_{f_1}^*, \ldots, x_{f_N}^*, y_{f_N}^*]^T.\end{aligned}$$

Or, using the state vector in the least squares SLAM formulation, the candidate solution is

$$V^f = [x_{r_1}^*, y_{r_1}^*, \phi_{r_1}^*, \ldots, x_{r_p}^*, y_{r_p}^*, \phi_{r_p}^*, x_{f_1}^*, y_{f_1}^*, \ldots, x_{f_N}^*, y_{f_N}^*]^T. \tag{3.47}$$

where

$$\phi_{r_i}^* = \text{atan2}(s_{r_i}^*, c_{r_i}^*),\ i = 1, \ldots, p. \tag{3.48}$$

The key question now is how close the candidate solution derived from this method to the optimal solution to the least squares SLAM problem. In [74] it has been proved that, for sensor network localization problem, when the problem is uniquely localizable [74], SDR solution is exactly the global optimal solution to the original problem.

Theorem 3.1: For the point feature SLAM problem (2.11) or (2.12), suppose all the measurements are accurate (no sensor noise), then the SDR result (3.47) is the optimal solution to the original SLAM problem (2.11) or (2.12).

Proof. Since all the measurements are accurate, the optimal solution to the least squares SLAM problem is the one that can achieve zero value in objective function and the covariance matrices and constraints will not affect the result. Thus, when all the measurements are accurate, the QCQP problem ((3.28) or (3.39)) is equivalent to the least squares SLAM problem ((2.11) or (2.12)).

When all the measurements are accurate, suppose ϑ is the optimal solution to the QCQP problem (3.40) (which is standard form of (3.28) or (3.39)), ϑ will make the value of objective function for the QCQP problem exactly zero. Because the objective function in QCQP problem only contain square terms, any $\tilde{V} \neq \vartheta$ will result in a greater value of objective function in (3.40) i.e.

$$\tilde{V}^T Q_0 \tilde{V} + a_0^T \tilde{V} + b_0 > 0.$$

From the semi-definite constraint of the SDP problem $X - \tilde{V}\tilde{V}^T \succeq 0$, we can have:

$$Tr(Q_0 X) + a_0^T \tilde{V} + b_0 \geq Tr(Q_0 \tilde{V}\tilde{V}^T) + a_0^T \tilde{V} + b_0. \tag{3.49}$$

Therefore, when $\tilde{V} \neq \vartheta$

$$Tr(Q_0 X) + a_0^T \tilde{V} + b_0 \geq \tilde{V}^T Q_0 \tilde{V} + a_0^T \tilde{V} + b_0 > 0. \tag{3.50}$$

However, as the objective function in SDP problem (3.45) is the same as the objective function in QCQP problem (3.40), the objective function value for SDP solution $\tilde{X}^*$ should be zero. Therefore $\tilde{V}^*$ from the SDP solution $\tilde{X}^*$ has to be equal to ϑ. □

This proves that the SDR result V^f in (3.47) is the optimal solution to least squares SLAM problem when all the measurements are accurate. However, in reality, measurements always contain noise and the candidate solution from the SDR result is only an approximate solution to the SLAM problem. Whenever optimal solution to the SLAM problem is desired, one can use the candidate solution as an initial guess and solve the least squares SLAM problem using local optimization approaches. Since the candidate

solution is close to the optimal solution as will be shown in Chapter 4, the local optimization algorithms such as Gauss-Newton can converge to the global optimal solution within a few iterations.

3.8 Summary

This chapter has proposed a convex optimization based approach for point feature SLAM problem. Through a new representation for the robot orientation and appropriate approximation for the covariance matrix, the point feature SLAM problem can be formulated as a QCQP problem. When applying SDR technique, the non-convex QCQP problem can be relaxed into SDP problem, which is a convex optimization problem. Unique solution to the SDP problem can be obtained without any initial state estimates. Unlike traditional LS method which does not guaranteed converge to global minimum, the proposed method only has one global minimum. As the proposed method convert LS problem into a convex optimization problem, the relaxed SLAM problem can be solved in polynomial time. Although the proposed method only solves the relaxed SLAM problem, it will be shown in the next chapter that, result from the proposed approach is very close to global minimum.

4 Evaluation for the Semi-definite Relaxation Based SLAM Approach

4.1 Introduction

In Chapter 3, a SDR based approach is proposed and unique solution can be obtained. It also has been shown in Chapter 3 that, when the measurement and odometry information is accurate, the candidate solution obtained from this approach is the optimal solution to the original least square problem. However, in reality, measurements always contain noise. It is necessary to evaluate the performance of the SDR based approach when noise presents. This chapter is designated to fulfil this task.

To evaluate the proposed SDR approach, two types of evaluation approaches are adopted: 1. evaluation using error covariance; 2. evaluation using value of objective function. Using the computer simulation and experimental result, it will be shown in this chapter that the SDR result is close to the optimal result obtained from the non-linear least squares.

The chapter is organized as follows. Section 4.2 describes the evaluation method used. Section 4.3 evaluates the SDR approach using simulation data. Section 4.4 evaluates the SDR approach using experimental data. Section 4.5 summarizes this chapter.

For all the experiments, SeDumi [75] is used as the SDP solver. Local optimization techniques are *NOT* applied to refine the candidate solution obtained through SDR in order to compare the candidate solution with the optimal solution.

4.2 Evaluation Methods

Because the proposed SDR based approach relaxes the least squares SLAM problem to a convex problem, the state estimate obtained from the proposed SDR approach does not contain any uncertainty information. Thus, direct comparison such as Normalized Estimation Error Squared (NEES) and 2 sigma confidence check may not be suitable for our case. Therefore, to compare the candidate solutions with the optimal solution to the least squares pose only SLAM problem, we adopt two types of evaluation approaches.

4.2.1 Error covariance of the algorithm

Although the SDR result does not provide the uncertainty of the estimate, the covariance matrix of estimation error of the algorithm can still be estimated [23] using Monte Carlo simulations. Let V_T be the ground truth of the state vector to be estimated. Let r be the number of Monte Carlo runs. For any particular estimation algorithm, let V_l be the estimate from the $l-$th run. Then the mean of all estimates of the algorithm can be obtained by

$$e_{MC} = \frac{1}{r}\sum_{l=1}^{r} V_l \tag{4.1}$$

and the covariance of the estimate error of the algorithm is

$$P_{MC} = \frac{1}{r}\sum_{l=1}^{r} (V_l - V_T)(V_l - V_T)^T \tag{4.2}$$

We compare the mean estimation error and the covariances of the proposed SDR approach with that of the optimal solution to the least squares (LS) SLAM problems. For comparison purpose, we assume that the LS result when ground truth is used as initial guess is the global optimal solution. By using this comparison method, all the dimension of the state estimate can be evaluated.

4.2.2 Value of objective function

To evaluate how close the candidate solution to the optimal solution, the value of objective function in (2.12) at the candidate solution V^f obtained from the proposed SDR

technique (3.47) is compared with the value of objective function at the global optimum V^u.

4.3 SLAM Result Using Simulation Data

4.3.1 Small scale simulation

A small simulation environment is used to perform a comparison of the error covariance of different algorithms. The $12m \times 12m$ environment contains 40 point features and the robot moved 72 steps. Each step, the robot moves 1 meter and rotates at a predefined angle. The true robot trajectory is shown in Fig. 4.1. The robot is assumed to have a sensor with 180 degree field of view and maximal range of 4 meters to acquire measurements of the relative position between robot and features. The observation and odometry information are corrupted with zero mean Gaussian noise. For observation noise, the standard deviation on x and y are $s * 0.1m$ and $s * 0.2m$ where s is a scale factor. The standard deviation of odometry noise on x, y and ϕ are $s * 0.1m$, $s * 0.2m$ and $s * 0.1rad$ respectively.

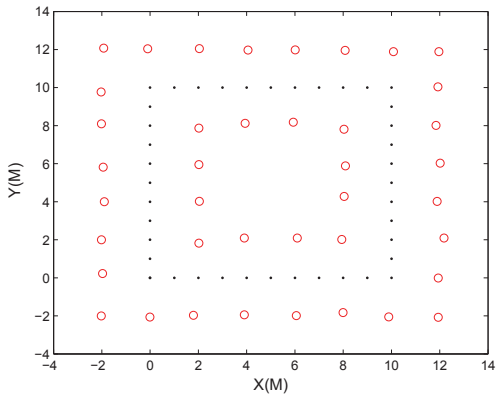


Figure 4.1: Simulation environment (red circles are beacon location; black dots are robot position).

We compare the error covariance at two noise levels (i.e. $s = 0.5$ and $s = 1$). At each noise level we generate 1000 datasets by Monte Carlo simulations and the comparison of error covariance P_{MC} are shown in Fig. 4.2(a) - Fig. 4.2(b). It can be observed

that, for both cases, the mean of the SDR approach is very close to the mean of the LS approach. Further more, the error covariance for the LS approach is bounded by the error covariance for the SDR approach.

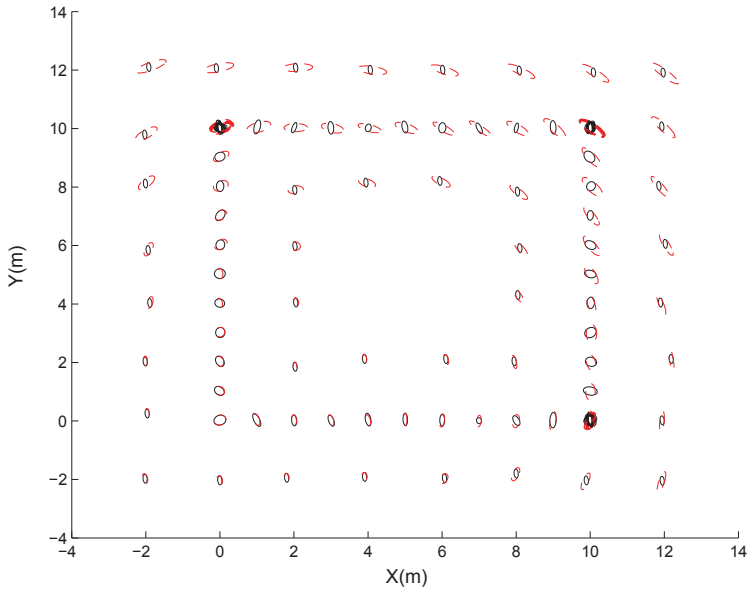
4.3.2 Larger scale simulation

The $25m \times 25m$ environment contains 76 point features and the robot moves 256 steps. The simulation environment is shown in Fig. 4.3(a). We compare the SDR result with optimal solution obtained through LS at various noise level (i.e. $s \geq 0$). At each noise level 20 datasets are generated to perform the evaluation.

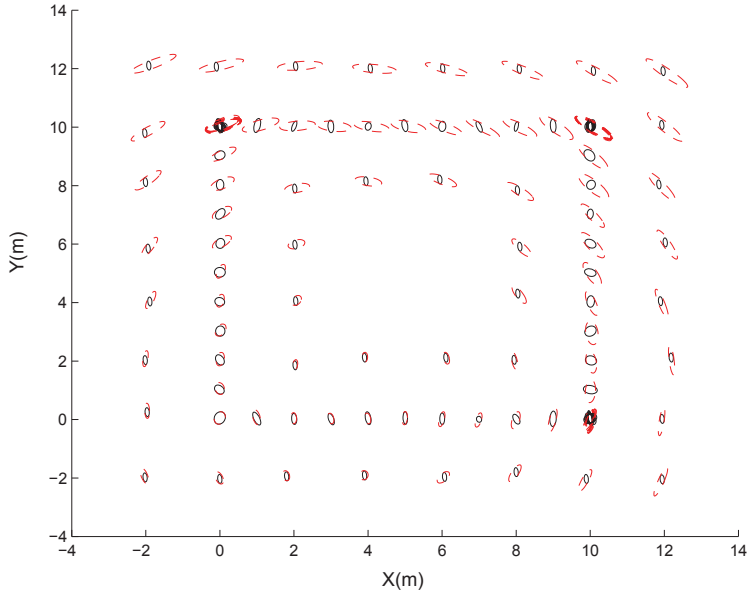
One odometry at noise level 1 (i.e. $s = 1$) is shown in Fig. 4.3(b). For this dataset, the difference between ground truth and the feature position estimate from SDR is compared with that of the difference from LS. The result is plotted in Fig. 4.3(c). The difference between ground truth and robot pose estimate from the two algorithms is shown in Fig. 4.3(d).

To evaluate the SDR performance when odometry is not available, odometry information is removed from the dataset then the SDR technique is applied on the observation data only. Fig. 4.3(e) shows the difference between ground truth and feature position estimate from SDR and LS. The difference between ground truth and robot pose estimate is shown in Fig. 4.3(f). It can be seen that, for both the odometry available and unavailable case, the SDR result and the optimal LS solution are similar.

For this case, the evaluation is performed by comparing the value of objective function in (2.11) and (2.12) at the candidate solution V^f obtained from the proposed SDR technique with the value of objective function at the global optimum. Due to the space limitation, only the results with the most significant difference between the value of objective function from SDR approach and the value of objective function at the global optimum among the 20 datasets at each noise level are shown. The result for odometry available case is tabulated in Table 4.1 and the result for odomety unavailable case is shown in Table 4.2. It can be seen that for both cases, the value of objective function for SDR is less than 110% of the optimal value.

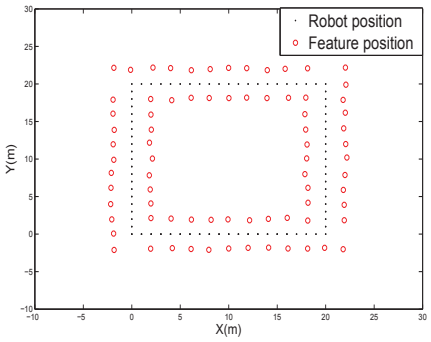


(a) The error covariance comparison of LS and the proposed SDR approach at noise level 0.5 (solid line is from LS; dashed line is from SDR approach).

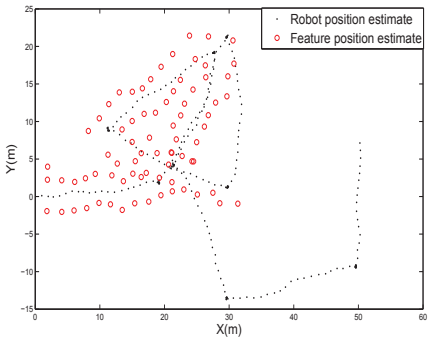


(b) The error covariance comparison of LS and the proposed SDR approach at noise level 1 (solid line is from LS; dashed line is from SDR approach).

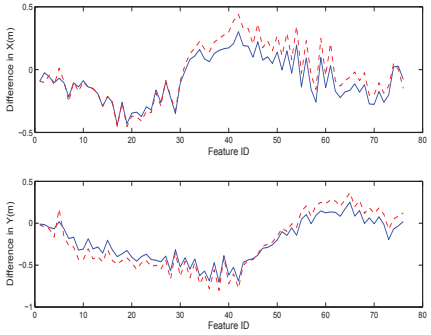
Figure 4.2: Results using small simulation data.



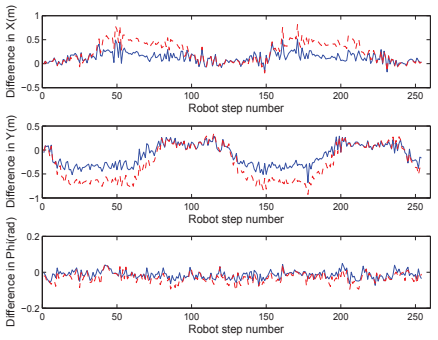
(a) Simulation environment.



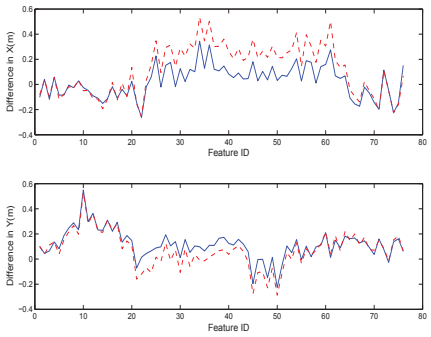
(b) One robot odometry example at noise level 1.



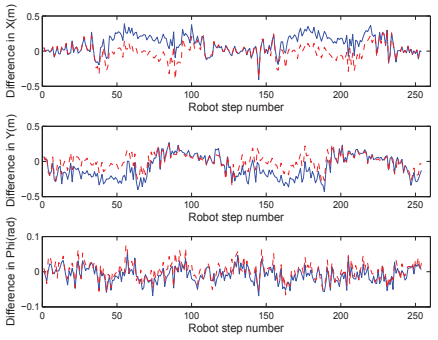
(c) Comparison of feature position estimate error from SDR and LS for odometry available case (solid line is from LS; dashed line is from SDR).



(d) Comparison of robot pose estimate error from SDR and LS for odometry available case (solid line is from LS; dashed line is from SDR).



(e) Comparison of feature position estimate error from SDR and LS for odometry unavailable case (solid line is from LS; dashed line is from SDR).



(f) Comparison of robot pose estimate error from SDR and LS for odometry unavailable case (solid line is from LS; dashed line is from SDR).

Figure 4.3: Results using simulation data.

Table 4.1: Value of Objective Function When Odometry Available

Noise level	Value of objective function for the SDR result	Value of objective function for the optimal solution	SDR/optimal
$s = 0$	0	0	-
$s = 1$	2863.7	2714.5	105.50%
$s = 2$	3002.3	2834.8	105.91%
$s = 5$	3208.8	2940.5	109.12%

Table 4.2: Value of Objective Function When Odometry Unavailable

Noise level	Value of objective function for the SDR result	Value of objective function for the optimal solution	SDR/optimal
$s = 0$	0	0	-
$s = 1$	2818.8	2705.1	104.74%
$s = 2$	2833.3	2684.9	104.99%
$s = 5$	2964.0	2738.2	108.25%

4.4 SLAM Result Using Experimental Data

To further validate the proposed algorithm, experiments have been performed on the Victoria Park dataset [30] and the DLR-Spatial Cognition dataset [44]. As the memory requirement for our current implementation is high, the current implementation is not able to process either of the two datasets at once. Thus, in order to process the two datasets, map joining method [35] has been used.

4.4.1 Result for the Victoria Park dataset

Odometry and observation data from every 100 robot steps has been used to build a submap. For the whole Victoria Park dataset, 69 submaps have been created. If the SDR result to each submap is refined using local optimization and then applied to map joining, the final result will be the same as the result of joining the optimal submaps. Therefore we present the map joining result using the submaps obtained by SDR without further refinement for comparison. The submap building and map joining process is described as follows. Firstly, SDR is applied on the submap data. Then the covariance matrix for the submap has been computed by evaluating the Jacobian at the candidate solution

obtained from the method described in Section 3.7. Finally, map joining algorithm is applied on all the local submaps. In this thesis, the I-SLSJF [35] technique, which is a local submap joining algorithm that fuses the local maps together using Extended Information Filter (EIF) and LS, is used.

For each submap the value of objective function from SDR result and the LS result is compared. As ground truth is not available for the experimental dataset, we treat LS solution when odometry is used as initial guess as the global optimal solution. Fig. 4.4 shows the value of objective function comparison for each submap for the Victoria Park dataset.

Fig. 4.5 shows final result comparison between the result from proposed method and result from LS of the whole dataset. The comparison suggests that, result obtained from the proposed method is almost identical with the result obtained from LS.

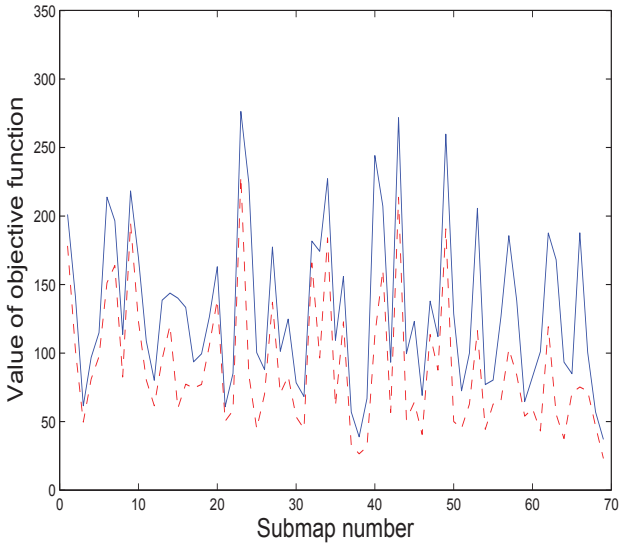


Figure 4.4: Comparison of the values of objective function for submaps of Victoria park dataset. Solid line is from SDR; dashed line is from LS

For this dataset, SDR result and the LS result has the most significant difference in value of objective function at submap 24. Fig. 4.6(a) shows the submap comparisons between LS result and SDR result for submap 24. The difference between feature location estimates for the two methods is shown in Fig. 4.6(c) and the difference between robot

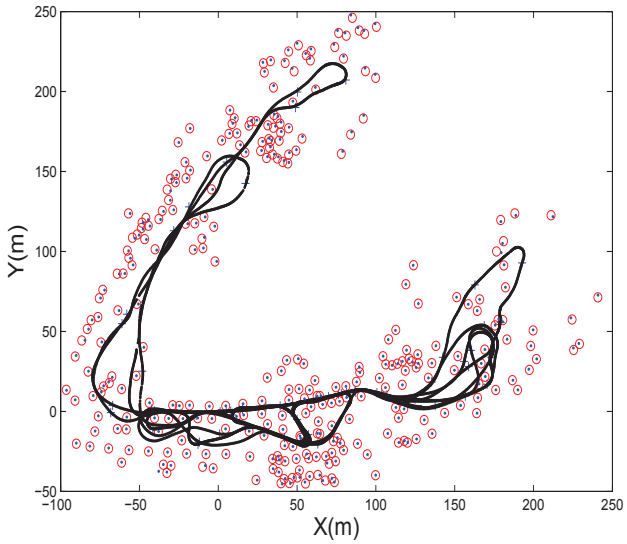


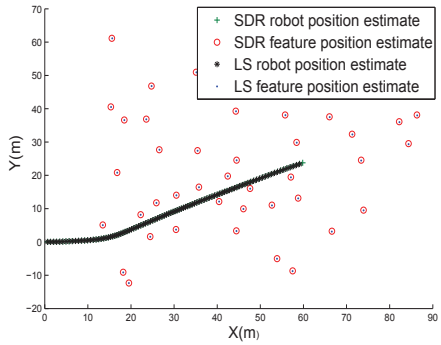
Figure 4.5: Final results comparison using the Victoria Park dataset. Circle: feature estimate using proposed SDR method with map joining, dot: feature estimate using LS.

pose estimates is shown in Fig. 4.6(e). At submap 67 the SDR result and the LS result has the least significant difference in value of objective function. Fig. 4.6(b) shows the submap comparisons between LS result and SDR result. The difference between feature location estimates for the two methods is shown in Fig. 4.6(d) and the difference between robot pose estimates is shown in Fig. 4.6(f).

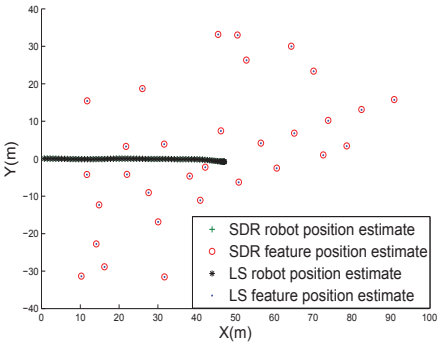
4.4.2 Result for the DLR-Spatial Cognition dataset

For DLR-Spatial Cognition dataset, odometry and observation data from every 100 robot steps has been used to build a submap. For this dataset altogether 33 submaps have been created. The objective function value comparison for each submap is plotted in Fig. 4.7. Fig. 4.8 shows final result comparison between result from the proposed method with map joining and result from LS of the whole dataset.

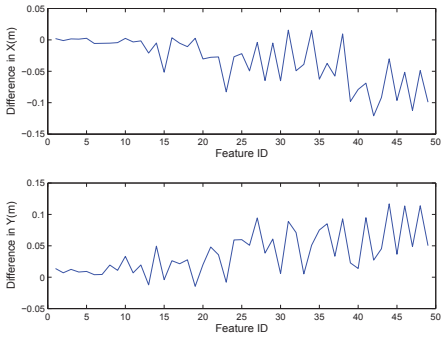
For this dataset, SDR result and the LS result has the most significant difference in value of objective function at submap 4. Fig. 4.9(a) shows the submap comparisons



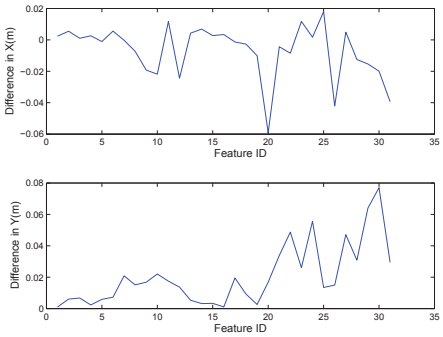
(a) SDR result and LS result for submap 24.



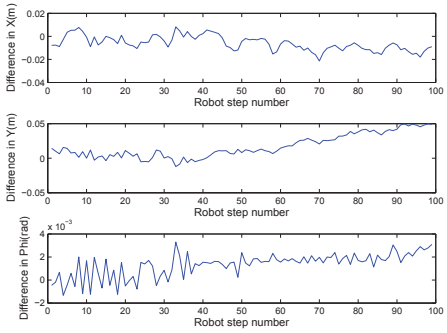
(b) SDR result and LS result for submap 67.



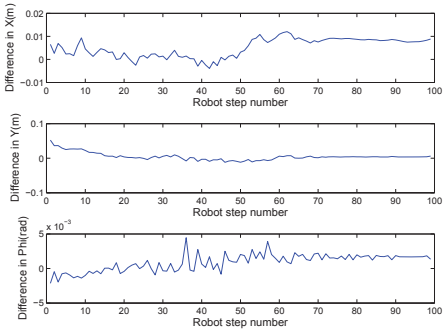
(c) Feature position difference between SDR result and LS result for submap 24.



(d) Feature position difference between SDR result and LS result for submap 67.



(e) Robot pose difference between SDR result and LS result for submap 24.



(f) Robot pose difference between SDR result and LS result for submap 67.

Figure 4.6: Submap result comparison for the Victoria Park dataset.

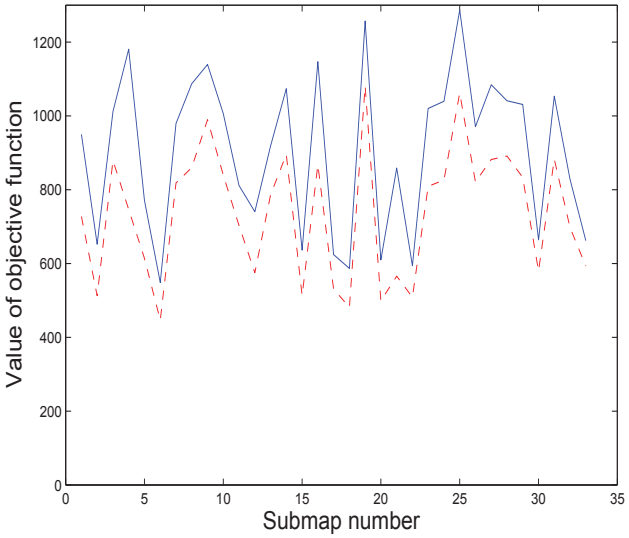


Figure 4.7: Comparison of the values of objective function for submaps of DLR-Spatial Cognition dataset. Solid line is from SDR and dashed line is from LS.

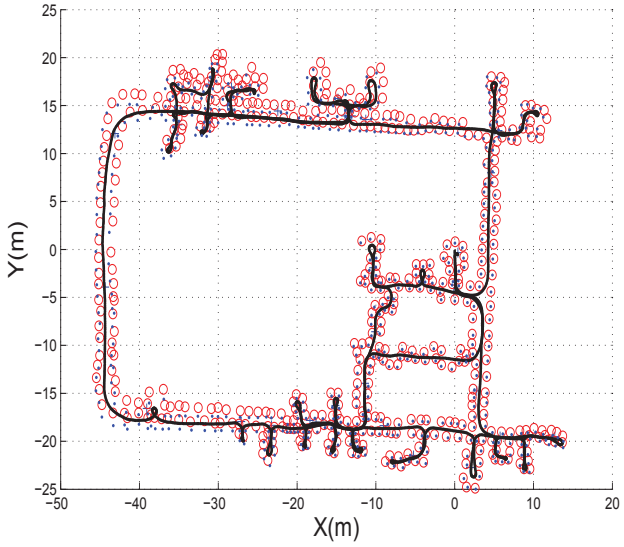
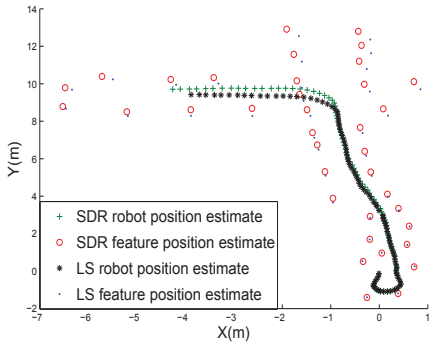


Figure 4.8: Final results comparison using the DLR dataset. Circle: feature estimate using proposed SDR method with map joining, dot: feature estimate using least squares.

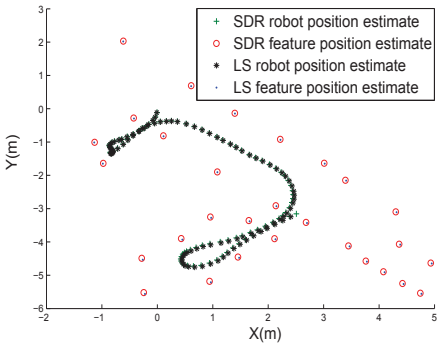
between LS result and SDR result for submap 4. The difference for feature estimate between the two methods is shown in Fig. 4.9(c) and the difference for robot pose estimate is shown in Fig. 4.9(e). At submap 30 the SDR result and the LS result has the least significant difference in value of objective fuction. Fig. 4.9(b) shows the submap comparisons between LS result and SDR result. The difference between feature location estimates for the two methods is shown in Fig. 4.9(d) and the difference between robot pose estimates is shown in Fig. 4.9(f).

4.5 Summary

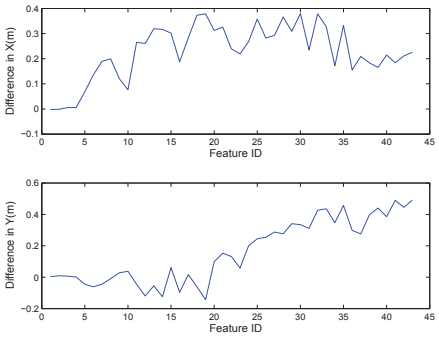
This chapter compares results obtained from proposed SDR approach and non-linear least square. Using simulation and experimental data results, it has been shown that the SDR result is very close to the global optimal solution to the SLAM problem and as such can either be directly used or exploited as an initial guess to the more traditional nonlinear least squares optimization algorithms if more accurate SLAM result with associated uncertainties are required.



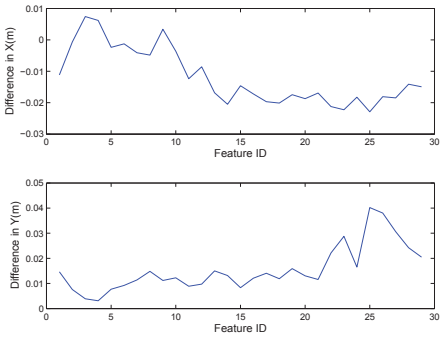
(a) SDR result and LS result for submap 4.



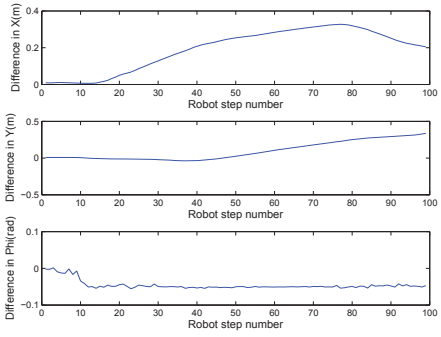
(b) SDR result and LS result for submap 30.



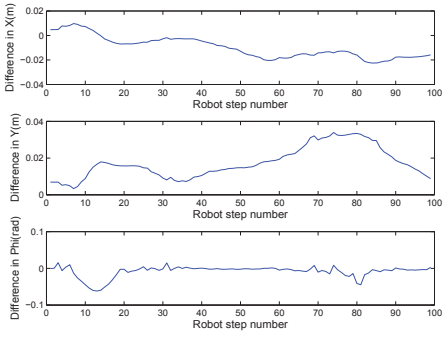
(c) Feature position difference between SDR result and LS result for submap 4.



(d) Feature position difference between SDR result and LS result for submap 30.



(e) Robot pose difference between SDR result and LS result for submap 4.



(f) Robot pose difference between SDR result and LS result for submap 30.

Figure 4.9: Submap result comparison for the DLR-Spatial Cognition dataset.

5 Curve Feature Based SLAM

5.1 Introduction

In Chapter 3, a convex optimization based approach has been proposed. We have also demonstrated in Chapter 4 that the proposed method is a good approximation of the original problem. However, as stated in Chapter 2, in point feature based SLAM, only a small portion of the information is used. To utilize more information, in particular when laser scan data is used, curve feature based SLAM is needed.

In this chapter, a general formulation for the 2D curve feature based SLAM problem is proposed. Different from other existing SLAM formulations, in the proposed SLAM formulation, the environment is represented by a set of continuous curves instead of point clouds or occupancy grids. The variables of the optimization problem are the robot poses as well as the parameters describing the curves. It has been shown that, point feature based SLAM is a special case of the proposed SLAM formulation.

In the new SLAM formulation, negative information which indicates the absence of an expected sensor reading is exploited as useful information. The event of not detecting an object is treated as evidence that can be used to update its probability density function [41]. It has been applied in the context of object tracking [70] and Markov localization [31]. It has also been mentioned in the context of simultaneous localization and mapping (SLAM) where it is used to adjust the confidence in landmark candidates [57].

As the objective function to be optimized contains discontinuities, the objective function can not be solved by standard gradient based approaches and thus a Genetic Algorithm (GA) based method is applied. Matching of laser scans acquired from relatively far apart robot poses is achieved by applying GA on top of the Iterative closest point (ICP) algorithm.

This chapter is organized as follows. Section 5.2 formulates the general 2D feature based SLAM problem where traditional point feature SLAM is a special case. Section 5.3 formulates the laser scan based SLAM optimization problem. Section 5.4 analyzes the properties of the proposed formulation. Section 5.5 proposes one approach to solve the optimization problem. In Section 5.6, implementation issues for two scans in real situation are discussed. Section 5.7 discusses the multi-scan SLAM implementation. Section 5.8 summarizes the chapter.

5.2 Curve Feature SLAM Formulation

When formulating the curve feature based SLAM problem, the environment is represented by a number of continuous curves and each curve has been treated as one feature. A continuous curve can be expressed by

$$\{(x, y) : x = x_c(t), y = y_c(t), \text{ for some } 0 \leq t \leq 1\} \tag{5.1}$$

where $x_c(t)$ and $y_c(t)$ are two continuous functions defined on $0 \leq t \leq 1$. For example, a line segment with end points (x_1, y_1) and (x_2, y_2) is a simple curve with

$$x_c(t) = tx_1 + (1-t)x_2,\ y_c(t) = ty_1 + (1-t)y_2,\ 0 \leq t \leq 1. \tag{5.2}$$

A B-Spline of order k with control points $(x_i, y_i),\ i = 0, \ldots, n$ is a curve with

$$x_c(t) = \sum_{i=0}^{n} x_i \beta_{i,k}(t),\ y_c(t) = \sum_{i=0}^{n} y_i \beta_{i,k}(t). \tag{5.3}$$

where $\beta_{i,k}(t)$ is given in (2.54). A single point feature (x_0, y_0) can be regarded as an even special curve where

$$x_c(t) \equiv x_0,\ y_c(t) \equiv y_0,\ 0 \leq t \leq 1.$$

The environment is assumed to contain N curves in total and curve j $(1 \leq j \leq N)$ can be expressed by

$$\{(x, y) : x = x_j(t), y = y_j(t), \text{ for some } 0 \leq t \leq 1\} \tag{5.4}$$

where $x_j(t)$ and $y_j(t)$ are two continuous functions defined on $0 \leq t \leq 1$.

The robot initial pose is set to $X_{r_0} = (0, 0, 0)$. Assuming the robot move $p-1$ steps and at each position X_{r_i} the robot gets k_i sensor readings $\{z_1^i, \cdots, z_{k_i}^i\}$ by observing the environments from its sensors the general feature based SLAM optimization problem aims to find the $2N$ functions

$$X_{f_j} = (x_j(t), y_j(t)),\ 0 \leq t \leq 1,\ j = 1, \cdots, N$$

and p robot poses

$$X_{r_i} = (x_{r_i}, y_{r_i}, \phi_{r_i}),\ 1 \leq i \leq p,$$

such that the observations and the odometry information are consistent with the map and the robot poses. Assuming the noise on the odometry information and the sensor readings are Gaussian, the objective can be achieved by minimizing the following least squares objective function

$$\sum_{i=0}^{p} \sum_{j=1}^{k_i} (\tilde{z}_j^i)^T P_{z_j^i}^{-1} \tilde{z}_j^i + \sum_{k=0}^{p-1} \tilde{o}_i^T P_{o_i}^{-1} \tilde{o}_i \tag{5.5}$$

where

$$\tilde{z}_j^i = z_j^i - H^{z_j^i}(X(t), Y(t), X_{r_i}, j) \tag{5.6}$$

$$\tilde{o}_i = o_i - H^{o_i}(X_{r_i}, X_{r_{i+1}}) \tag{5.7}$$

Here, z_j^i is the j-th sensor reading from robot pose i. The corresponding theoretical sensor reading is denoted by $H^{z_j^i}(X(t), Y(t), X_{r_i}, j)$ provided that the curves are given by $X(t) = x_1(t), \cdots, x_N(t), Y(t) = y_1(t), \cdots, y_N(t)$. It depends on the sensor used. For the case of laser range finder, the theoretical sensor reading is the distance between the robot pose and the first ray-feature intersection. The difference between the theoretical sensor reading $H^{z_j^i}(X(t), Y(t), X_{r_i}, j)$ and the actual sensor reading z_j^i is denoted by $\tilde{z}_j^i$. Matrix $P_{z_j^i}$ is the covariance matrix for z_j^i.The robot pose i is given by X_{r_i}. The relative pose information between pose i and pose $i+1$ is described by odometry o_i, the corresponding theoretical odometry is denoted by $H^{o_i}(X_{r_i}, X_{r_{i+1}})$ and $\tilde{o}_i$ is the difference between the theoretical odometry $H^{o_i}(X_{r_i}, X_{r_{i+1}})$ and the actual odometry o_i. Matrix P_{o_i} is the covariance matrix for o_i.

In the above formulations, each curve is expressed by infinite number of parameters. In practice, finite number of parameters (such as polynomials or B-Splines) is generally enough for accurately expressing 2D curves.

For SLAM problem, the sensor reading z_j^i means some robot-to-environment information. It can be presented in various forms. Here we discuss a special case

Special case: point feature SLAM

Consider a special case where the environment is formed by N points

$$(X, Y) = \{(x_j, y_j),\ j = 1, \cdots, N\}$$

$$x_j(t) \equiv x_j, y_j(t) \equiv y_j,\ 0 \leq t \leq 1,\ j = 1, \cdots, N.$$

Assuming all the N points can be observed from each pose, we have $k_i = N$ for $0 \leq i \leq p$, and for each $1 \leq j \leq N$. The sensor readings from each pose X_{rk} are the range and bearing measurement from the robot pose to the N points and the theoretical measurement can be expressed as

$$H^{z_j^i}(X(t), Y(t), X_{r_i}, j) = \begin{bmatrix} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \\ atan(\frac{y_j - y_i}{x_j - x_i}) - \theta_i \end{bmatrix}. \tag{5.8}$$

The feature based SLAM optimization problem (5.5) becomes the point feature range-bearing SLAM optimization problem formulation. This problem is a very clear least squares optimization problem [5] and has been well studied.

5.3 Curve Feature SLAM When Laser Scanner Used

In this section, using laser range finder as an example, we discuss how to apply the proposed formulation in more practical situation.

Consider a laser scanner that has the following configuration: the field of view of laser scanner is $[0^o, 180^o]$ and the resolution of the laser scanner is 1^o. The curve feature SLAM

optimization in this case is

$$\sum_{i=0}^{p}\sum_{j=1}^{181}(\tilde{z}_j^i)^T P_{z_j^i}^{-1}\tilde{z}_j^i + \sum_{i=0}^{p-1}\tilde{o}_i^T P_{o_i}^{-1}\tilde{o}_i \tag{5.9}$$

When odometry information is unavailable, the curve feature SLAM optimization problem becomes

$$\sum_{i=0}^{p}\sum_{j=1}^{181}(\tilde{z}_j^i)^T P_{z_j^i}^{-1}\tilde{z}_j^i. \tag{5.10}$$

Using this formulation, all the 181 laser readings have been included in the objective function. Even if the reading gives a maximal range, it is still compared with the theoretical reading.

Notice that, this formulation aims to provide estimate for all the curves in the environment: $X(t)$ and $Y(t)$, as well as the robot poses X_{r_i}. To explain this formulation, the following example is used.

Example 5.1: One line segment, robot moves one step

Assume the environment only contains one line segment with end points denoted by (x_1, y_1) and (x_2, y_2) (Fig. 5.1):

$$x = tx_1 + (1-t)x_2,\ y = ty_1 + (1-t)y_2,\ 0 \le t \le 1 \tag{5.11}$$

The robot started from $[0, 0, 0]$ and only moved once to X_{r_1}. The objective of our formulation is to get the parameters for the line segment (x_1, y_1) and (x_2, y_2) as well as the robot position $X_{r_1} = (x_{r_1}, y_{r_1}, \phi_{r_1})$ from the two laser scans. So there are 7 variables in this optimization problem.

5.4 Special Properties of the Objective Function

In Section 5.3, the curve feature SLAM formulation when laser scanner is used has been derived. All the available information is included in the formulation. Continue using the line segment example (Example 5.1), some properties of this formulation are discussed.

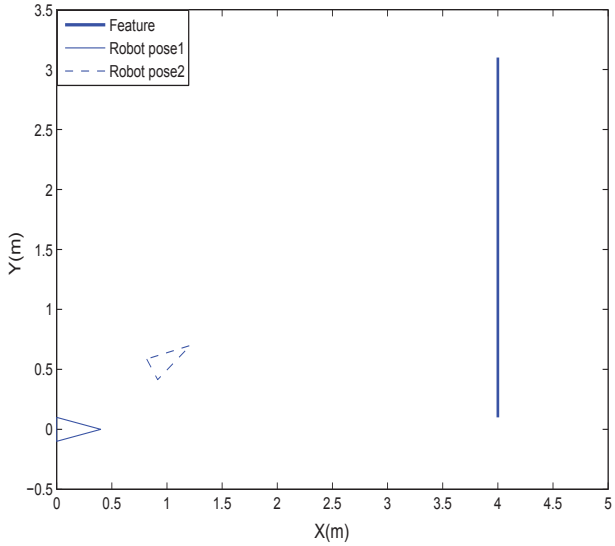


Figure 5.1: Example 5.1: One line segment, robot moves one step

5.4.1 Discontinuity of the function

Suppose the parameters for the line segment ((x_1, y_1) and (x_2, y_2)) and the robot's position in x-axis x_{r_1} are given. By varying the remaining two variables y_{r_1} and ϕ_{r_1}, the function plot of (5.9) for this case can be generated. The plot is shown in Fig. 5.2.

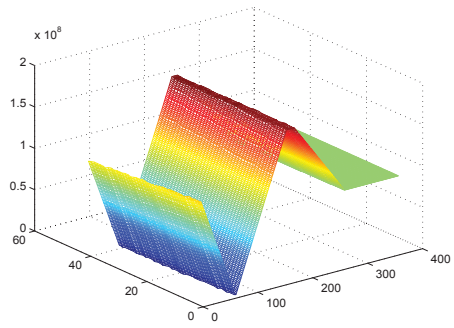
From Fig. 5.2(c), it can be clearly seen that (5.9) is a discontinuous function.

There are two reasons for this phenomenon.

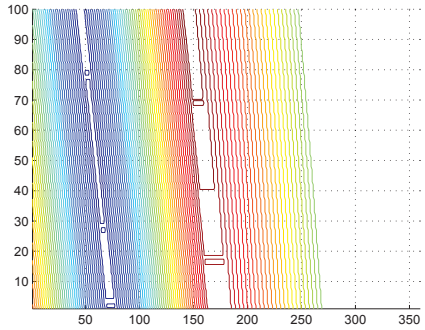
1) Including the maximal laser range in the objective function.

2) Rigorous sensor reading comparison which compares each individual laser reading with the theoretical reading.

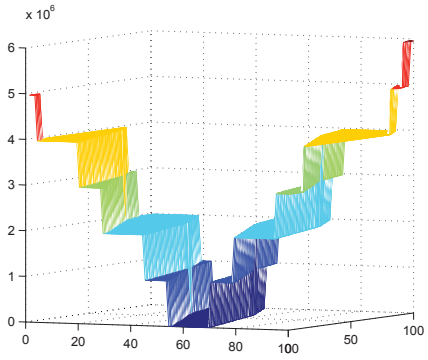
We continue explain this phenomenon using the line segment case. In this case, the true measurement is shown in Fig. 5.3(a). From Fig.5.3(b) - Fig.5.3(d) it can be observed that, a little change in y_{r_1} or θ_{r_1} dramatically enlarges the value of the objective function however large change in x_{r_1} does not change the value of objective function much. This is due to the occurrence of theoretical laser reading where maximal range actually occurs. This phenomenon further proves that the laser's maximal range do carry important



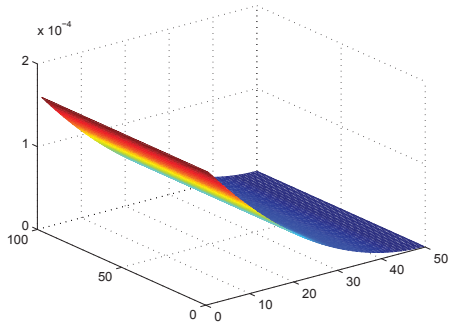
(a) The objective function plot of the optimization problem.



(b) The contour plot of the objective function with 50 layers



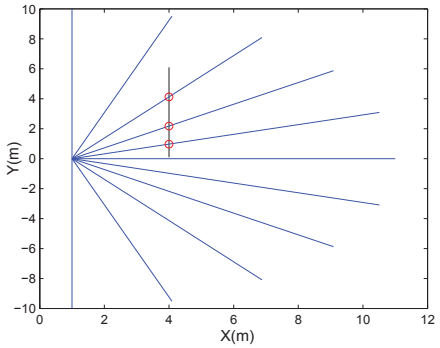
(c) Zoomed objective function plot. This plot is generated by varying y_{r_1} within $\pm$2cm and ϕ_{r_1} within $\pm$2 degrees.



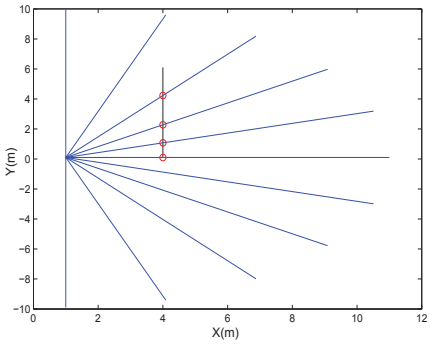
(d) Further zoomed objective function. This plot is generated by varying y_{r_1} within $\pm$1cm and ϕ_{r_1} within $\pm$0.1 degrees. The global minimum in this case is a line segment which defined by the laser resolution

Figure 5.2: Function plot of Example 5.1: all the other variables are fixed apart from y_{r_1} and θ_{r_1}

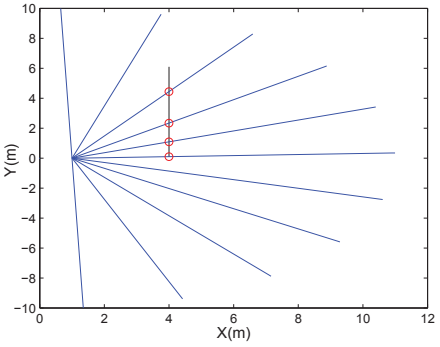
information.



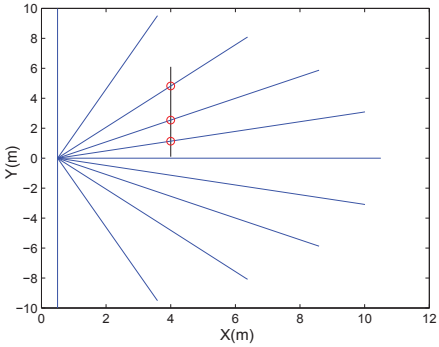
(a) At true location the laser has 3 rays intersect with the feature



(b) Change the robot's position in y-axis by 1cm results 4 laser rays intersect with the feature



(c) Change the robot's orientation by 2 degrees results 4 laser rays intersect with the feature



(d) Change the robot's position in x-axis by 0.5m results 3 laser rays intersect with the feature.

Figure 5.3: Explanation of the phenomenon

5.4.2 Local minimum

From Fig.5.2(c), it can also be noted that, the function has lots of local minima. And the objective function is constant in some small regions. Most of the existing SLAM methods will be trapped into such a local minimum because of their linearization nature.

5.5 One Approach for the Proposed SLAM Formulation

As discussed in Section 5.4, (5.10) is a discontinuous function, using traditional local optimization method is not appropriate. It is necessary to use global optimization methods to solve this problem. In this chapter, a Genetic Algorithm (GA) based approach is proposed to solve the curve feature based SLAM problem.

5.5.1 Increasing the chance by relaxation

Because of the discontinuity around global minimum, it may be not possible for GA to find the global minimum easily.

To overcome this, the *trimmed least squares* [69] technique is applied to smooth the objective function (5.10). That is, to ignore a few largest errors in the function (5.10). This actually means that the possibility that the laser sensor can sometimes generate a certain number of false readings has been taken into consideration.

In particular, we first compute all the $\tilde{z}_j^i$ and then ignore the largest $n_s * N_{threshold}$ terms among all the $(m+1) \times 181$ terms. Here n_s is the total number of segments in the laser scans and $N_{threshold}$ is the number of false laser reading we can expect from each segment (we choose $N_{threshold} = 2$ in our experiment since it is quite possible that the laser reading around the two end points of the curve may have large errors).

Utilizing the proposed relaxation method, we have changed the properties of the function. The threshold $N_{threshold}$ represents a trade-off between accuracy and convergence. Larger threshold result in larger chance to find the global minimum for the relaxed (5.10). However, instead of a single point, the global minimum of the relaxed (5.10) is a small region.

5.5.2 Theoretical measurement calculation for complex features

For the laser scan based SLAM problem, the theoretical measurement is the range reading in the direction of each individual laser ray. For complex features such as high order polynomials, the measurement does not have closed-form solution. In this situation, numerical method can be used to find the solution. Taken B-Splines as an example, the

theoretical measurement calculation process is discussed below.

Suppose at robot pose $X_r = (x_r, y_r, \phi_r)$, the robot observes previously observed B-Spline feature $X_f = (x(t), y(t)),\ 0 \leq t \leq 1$. The resolution of the equipped laser scanner is 1^o and the field of view is $[0^o, 180^o]$. The theoretical measurement for $j-$th laser ray can be derived by find the intersection between the laser ray and the B-Spline. This can be achieved by using the following steps.

Firstly, an orthonormal reference frame $X_J = [x_{cj}, y_{cj}, \phi_{cj}]$ has been defined. The coordinate system X_J is centered in the robot reference frame

$$x_{cj} = x_r,\ y_{cj} = y_r.$$

The orientation of the reference frame ϕ_{cj} is defined by the laser beam direction and robot orientation which is

$$\phi_{cj} = \phi_r + j. \tag{5.12}$$

Then the B-Spline need to be transferred to the coordinate system X_J and becomes $X_f^J = (x^J(t), y^J(t)),\ 0 \leq t \leq 1$. Using the affine invariance property of B-Spline, the B-Spline X_f can be transferred to X_J by transferring the control points of X_f. Let $X_i = [x_i, y_i],\ (i = 0, ..., n)$ be the control points for B-Spline X_f in global coordinate and $X_i^J = [x_i^J, y_i^J],\ (i = 0, ..., n)$ be the control points in coordinate X_J, the relationship between X_i and X_i^J is

$$\begin{bmatrix} x_i^J \\ y_i^J \end{bmatrix} = \begin{bmatrix} \cos\phi_{cj} & \sin\phi_{cj} \\ -\sin\phi_{cj} & \cos\phi_{cj} \end{bmatrix} \begin{bmatrix} x_i - x_{cj} \\ y_i - y_{cj} \end{bmatrix} \tag{5.13}$$

When transferring the B-Spline to the coordinate system X_J, the intersection between the laser ray and the B-Spline can be derived by finding t which makes $y^J(t) = 0$ and the theoretical measurement becomes $x^J(t)$. Figure 5.4 depicted this process. Since B-Splines are differentiable, numerical method such as Newton-Raphson method can be used to find the solution.

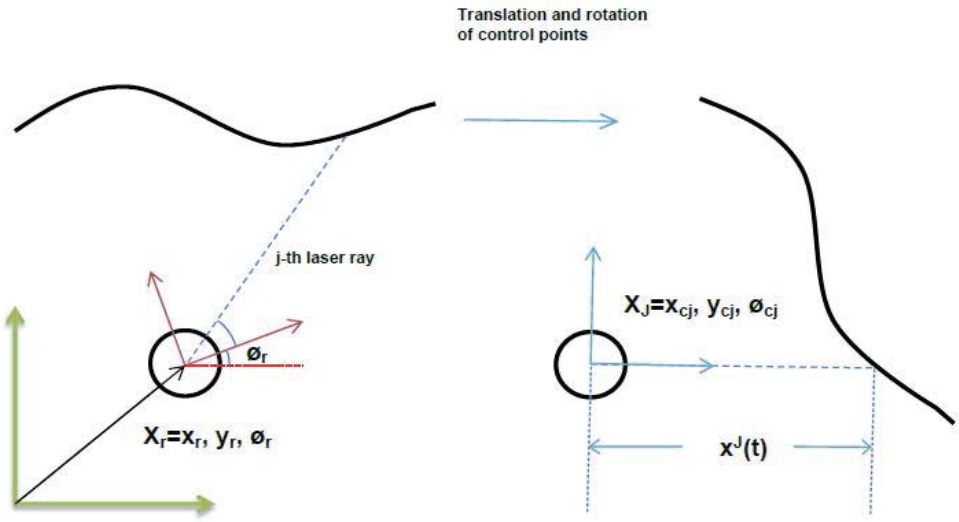


Figure 5.4: Theoretical measurement calculation for B-Spline features.

5.6 Implementation Issues - Optimization for Two Scans

In practice, when solving the SLAM problem using all the available scan, there will be many parameters to be estimated and the search space becomes too large. To simplify the problem, we firstly consider the situation where the robot only moved one step and acquired only two scans. In this section, we discuss some implementation issues for this situation. For the problem to be solvable, it is assumed that the two laser scans have more than $p_{overlap}$ percent overlap.

5.6.1 Limitations of the SLAM formulation and algorithm

Unknown feature number and feature type

In the laser scan SLAM optimization problem formulation (5.9), it is assumed that the total number and the type of each feature are known and their parameters are used as variables in the optimization problem. In practice, this information is not available but need to be worked out from the scans.

Along with this, the formulation for the expected measurement $H^{z_j^i}(X(t), Y(t), X_{r_i}, j)$ is not available without knowing the type of feature.

Infinite search space

Although in most cases the search space is bounded by odometry information, there are situations where odometry information is not available or very poor (e.g. loop closure). In this case, it will be challenging to get the global minimum from the infinite search space.

5.6.2 Problem of relating two scans

Suppose there are two scans available and the odometry information is not available and noise on all the measurements are Gaussian, the curve feature based problem is to find the relative pose X_{r_1} and the number and type of features involved in the two scans. This can be achieved by finding X_{r_1} to minimize

$$\sum_{j=1}^{181}[z_j^1 - H^{z_j^1}(X_1(t), Y_1(t), X_{r_1}, j)]^2 P_{z_j^1}^{-1} + \sum_{j=1}^{181}[z_j^0 - H^{z_j^0}(X_0(t), Y_0(t), j)]^2 P_{z_j^0}^{-1} \quad (5.14)$$

subject to the condition that the overlap between $(X_1(t),\ Y_1(t))$ and $(X_0(t),\ Y_0(t))$ is more than $p_{overlap}$. Here $(X_0(t),\ Y_0(t))$ represents the features in scan 0 and $(X_1(t),\ Y_1(t))$ represents the features in scan 1 (transferred into the coordinate of X_{r_0} using X_{r_1}). For this problem, these features are handled in implicit form (details of this handling is discussed below).

This problem is a scan matching problem without initial guess. Here we use an objective function (5.14) and the overlap constraint to formulate the problem as an optimization problem.

After the two scans are related to each other, the two scans can be transferred into one coordinate frame using X_{r_1}. Then to work out the total number and the type of features in the environment is not too difficult.

5.6.3 Approach for relating two scans

In this chapter, the following procedure is proposed to solve the scan relating problem.

Segmentation

Before proceed to optimization, the formula for expected measurement need to be decided. Due to the lack of environment knowledge, we perform segmentation on the laser scan and assume that each segment is corresponding to a continuous curve.

For our segmentation method, only two cases have been considered: (a) Points separated by maximal range; (b) Sudden change in consecutive points.

Use the merit of ICP

Directly using GA and try to find X_{r_1} is computationally very expensive to solve the scan relating problem because of the infinite search bound. Here we propose to use the advantages of ICP algorithm to speed up the optimization process. It is known that ICP always converge but may converge to a local minimum. Since the global minimum must be one of the local minima, we propose to use GA+ICP to find the solution quickly. The idea is to generate a set of random initial values (chromosomes), and then apply ICP to find the local minima around each of the initial values that satisfy the overlap constraint. The objective function (5.14) is then used to evaluate the local minima and get the fitness of each chromosome.

Evaluating the objective function of ICP result

For each local minimum obtained by ICP, the value of objective function is evaluated. At this stage, there are only two scans need to be aligned and ICP has already found the overlap area between the two scans, the computation of the objective function (5.14) becomes possible.

For the uncommon area of the two scans, there is no difference between the true measurement and the expected measurement. Thus only the difference for the common area identified by ICP need to be computed. In hybrid GA-ICP [53], the authors uses

sum of distances between matched pairs divided by the total number of matching pairs as fitness function. Different from their approach, in our implementation, we describe the common area in an implicit way. That is we use points from each segment to form a polyline to approximate the curve. The theoretical distance is approximated using ray tracing to polyline. As the laser scanner is sampled in a fine resolution, we consider this is a valid approximation.

After applying GA on top of ICP, the relative pose information between scans is derived.

5.6.4 Result

To test the proposed approach, the Intel dataset from the Robotics Data Repository [38] is used. In Fig. 5.5(d) it is shown that, the proposed approach is able to align two far apart laser scans without odometry information. The performance is better than vanilla ICP proposed in [7] and hybrid GA-ICP [53].

Using only 47 scans from the Intel dataset, we get the result shown in Fig. 5.6. To ensure enough overlap, when robot having straight motion, we selected approximately every 100 scans, and when robot is turning, we select around every 30 scans.

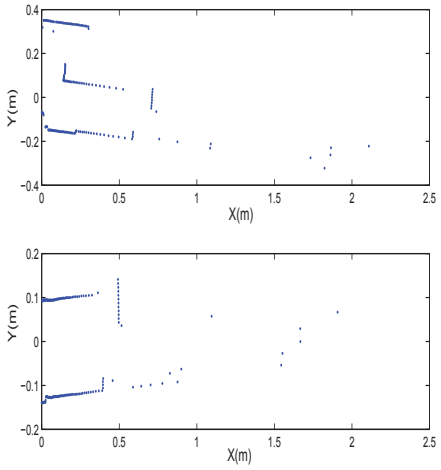
5.7 Multi-scan SLAM Implementation

Now we consider a more practical scenario when we have a number of scans and want to build a map.

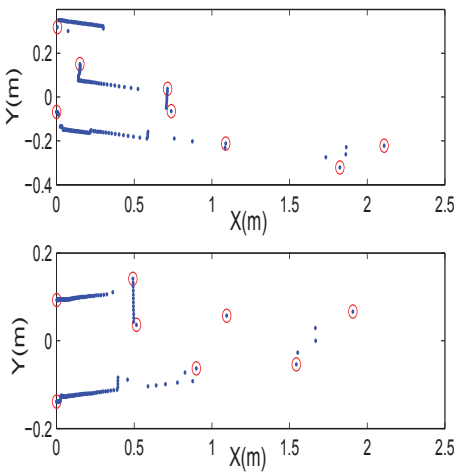
5.7.1 Multi-scan SLAM problem formulation

We assume there are $m + 1$ laser scans sampled from the $m + 1$ robot poses. We also assume there is no or very poor odometry information and we want to use the scans to build a map.

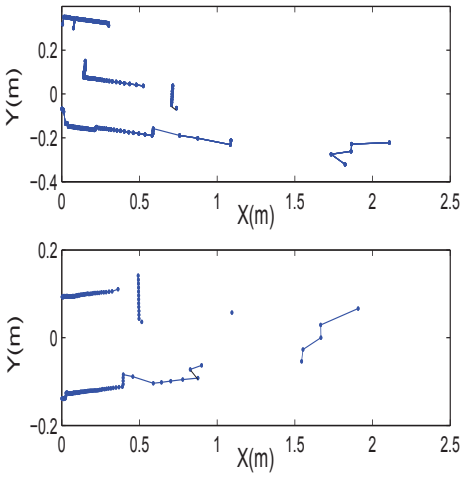
To solve any SLAM problem, the necessary condition is: there is mutual information between robot poses. The laser scan SLAM problem is no exception. It requires mutual information between scans. If there is no mutual information among scans, the solution



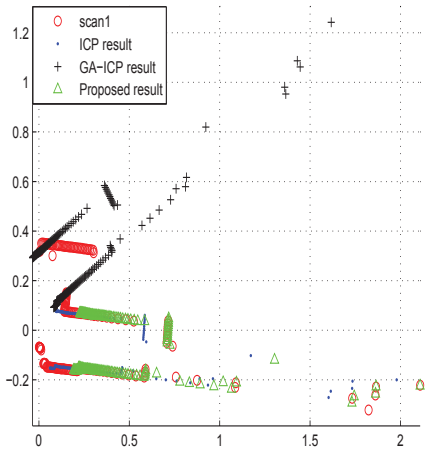
(a) Two raw scans from Intel lab data



(b) Scans after segmentation. Each segment is bounded by red circle



(c) For each segment, a polyline is formed to approximate the feature.



(d) Comparison of ICP, hybrid GA-ICP [53] and the proposed method. Proposed method is able to align two far apart laser scans without odometry

Figure 5.5: Align two far apart scans using the proposed method

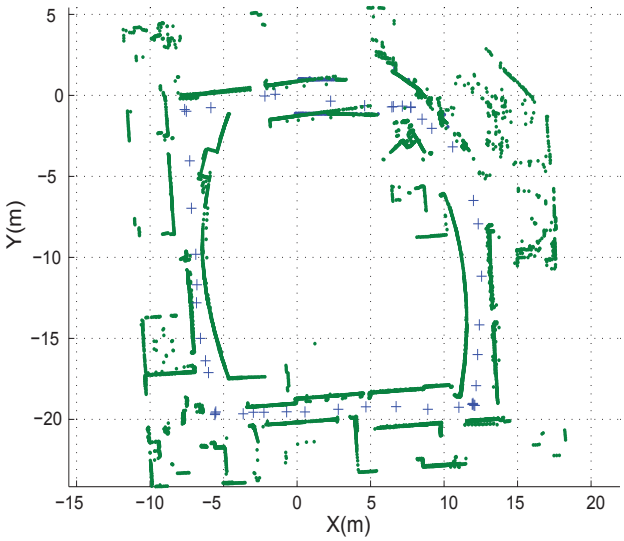


Figure 5.6: Result for deriving relative pose from far apart scans. This figure shows the result using 47 scans from the Intel dataset.

is not unique. Therefore, it is assumed that, if $m+1$ scans have been sampled in total, any scan S from the $m+1$ scans need to be directly or indirectly connected with the remaining m scans.

Normally, there are two situations where two scans will have overlap: (a) Two scans are sampled consecutively; (b) Two scans happened when loop is closed.

Multi-scan SLAM problem: Suppose there are $m+1$ scans available, find the number of features, type and parameters of each feature, and optimize the robot poses $X_{r_i},\ i = 1, \cdots, p$ and the feature parameters by minimizing (5.10).

Use GA+ICP to get the relative poses

From the previous section, we can solve the scan relating problem for two scans and get the relative pose. We perform this for all the scan pairs that have overlaps.

Use least squares to further reduce search space

As mentioned earlier, our method improves the convergence. However, because the use of "trimmed least square", the solution may not be the "global minimum". Before

solving the multi step SLAM problem using GA, we use linear least squares to smooth the trajectory and further reduce the search space.

Initialize the features

From the previous step, relative pose between scans had been established and each scan has been divided into segments. By relating segments between scans, we can get all the points of one feature. By curve fitting, we can get the initial estimate of the curve.

Final optimization

From Section 5.5, we can solve the SLAM problem by global optimization when the numbers, types, and parameter bounds are given. At this stage, the feature estimate and the robot pose estimate have been derived. The search space has been controlled in a small region. With small search space, the global optimization now becomes tractable. Applying GA and relaxation approach with a small search space we can get the feature and pose estimate which minimizes (5.10).

5.7.2 Results using Intel dataset

Using the least square smoothing, the robot trajectory has been improved. However, the result is not optimal. From Fig. 5.7(b), it can be observed that there are some misalignment at loop closure.

After applying GA to multi-step SLAM optimization problem, we get the result in Fig. 5.8. The misalignment at loop close has been fixed. Further more, instead of an occupancy grid map, the result map is a feature map.

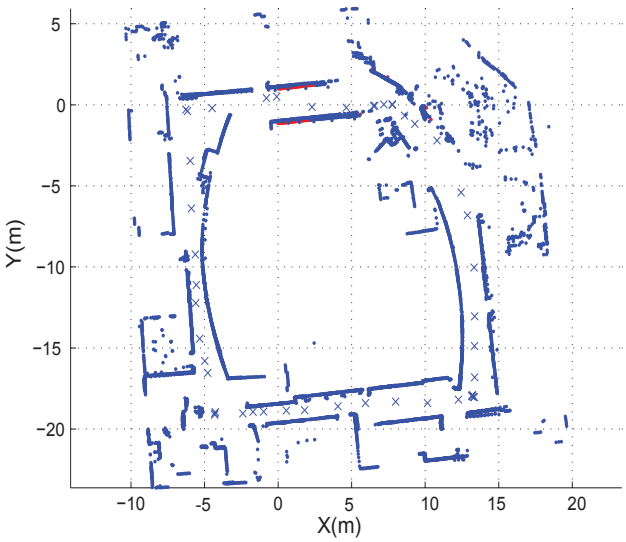
5.8 Summary

This chapter formulated the general feature based SLAM with laser data as an optimization problem. The problem formulation aims to maximize the information use. Both the bearing angle information and maximal range information is exploited. The new formulation is a generalization of point feature based least squares SLAM formulation.

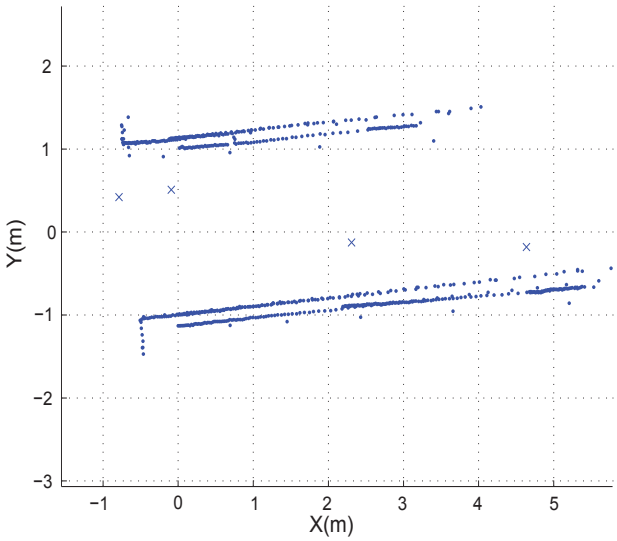
Because the objective function is not continuous and contains a lot of local minima. It is proposed to use some relaxations techniques such that it is tractable by global

optimization algorithms. The simulation and experimental results show that limited number of scans can be used to build up a map without odometry information by using the proposed relaxation and optimization techniques.

However, one problem can not be avoided is the convergence issue of the new formulation. As the new formulation involves discontinuity, the new formulation is hard to solve compare with point feature based least squares SLAM formulation. In next chapter, a conversion method from the curve feature SLAM to the point feature SLAM is proposed.



(a) Compare with Fig.5.6, the result is improved by using least square smoothing.



(b) Zoomed at the loop closure part.

Figure 5.7: The result after applying least square smoothing

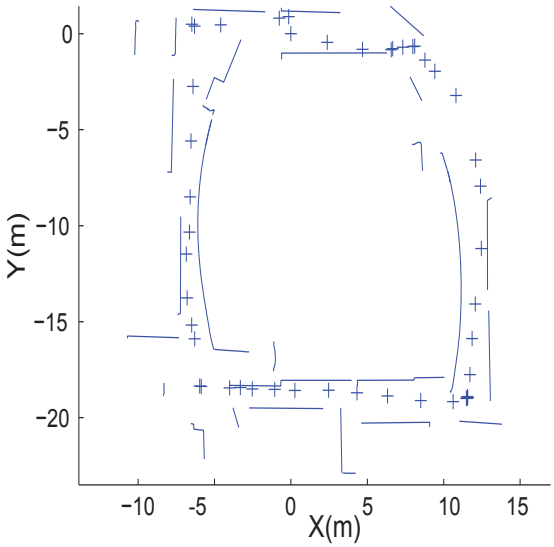


Figure 5.8: Multi step SLAM result using GA

6 Convert Curve Feature SLAM to Point Feature SLAM

6.1 Introduction

In the previous chapter, a general curve feature based formulation is proposed. However, it has been observed that, the formulation is hard to solve due to the discontinuity of the objective function.

In this chapter, it is proposed to transfer curve feature based SLAM to point feature based SLAM using B-Splines as features. Unlike the BS-SLAM proposed in [64], the observation model used in this chapter is a function of relative positions between the control points of the observed spline and the observation point. The uncertainty of the control points has been derived analytically from two sources: spline fitting error and the chord length error. It has been considered that the spline fitting error and the chord length error are not independent. Unlike the observation model in [64], optimization based techniques can be directly applied to the new observation model. Using this new observation model, the curve feature based SLAM problem can also be solved by the proposed SDR based approach.

This chapter is organized as follows. Section 6.2 proposes the new observation model. In Section 6.3, the data association method for the new observation model is described. Section 6.4 provides some consistency analysis on the new observation model. Section 6.5 shows results on B-spline SLAM using both simulation and experimental data. Section 6.6 summarizes the chapter.

6.2 The New Observation Model

6.2.1 Curve length parameterizations

It has been observed in Section 2.7.3 that, given the order, the knot vector and the control points, the location and the shape of the B-spline is completely decided. However, when using spline fitting to compute the control points, even when the order and the knot vector are all given, there are still infinite possible sets of control points that fit the points due to the infinite number of possible time sequence $r_v = [t_1, ..., t_m]$. Only when the order, the knot vector and the time sequence are all given, the set of control points can be uniquely determined. In the following we show how to decide the time sequence such that the same set of control points can be estimated from the scan data obtained from different observation points.

For the proposed observation model, the time parameter sequence need to be invariant to the observation point. As length of a spline is invariant with observation point, chord length method [52] is used. The chord length method uses the ratio between the cumulated chord length and the total chord length to approximate the time parameter t_j:

$$\begin{aligned} &l_1 = 0 \\ &l_j = \textstyle\sum_{i=1}^{j-1} \|d_{i+1} - d_i\| \\ &l_{m-1} = \textstyle\sum_{i=1}^{m-1} \|d_{i+1} - d_i\| \\ &t_j = l_j / l_{m-1} \end{aligned} \tag{6.1}$$

where d_i is the $i-$th data point and $\|\cdot\|$ is the Euclidean norm.

Once the time sequence is derived, the estimated control points can be simply obtained by spline fitting.

6.2.2 Covariance matrix derivation

To use the estimated control points as observation model to SLAM algorithm, the covariance matrix need to be correctly modelled. Assuming the uncertainty of the raw

measurement is S, the covariance matrix of control points can be derived from (2.59):

$$P_s = \frac{\partial \Phi M}{\partial M} S \frac{\partial \Phi M}{\partial M}^T \tag{6.2}$$

Since the size of M is $m \times 2$ and size of Φ is $n \times m$, by the product rule for matrix calculus [52] we have:

$$\frac{\partial \Phi M}{\partial M} = (M^T \otimes I_n)\frac{\partial \Phi}{\partial M} + (I_2 \otimes \Phi)\frac{\partial M}{\partial M} \tag{6.3}$$

where

$$\frac{\partial \Phi}{\partial M} = \frac{\partial \Phi}{\partial B}\frac{\partial B}{\partial M} = \frac{\partial (B^T B)^{-1} B^T}{\partial B}\frac{\partial B}{\partial M} \tag{6.4}$$

and $\otimes$ is the tensor product. Apply product rule on equation (6.4):

$$\frac{\partial \Phi}{\partial M} = ((B \otimes I_n)\frac{\partial (B^T B)^{-1}}{\partial B} + (I_m \otimes (B^T B)^{-1})\frac{\partial B^T}{\partial B})\frac{\partial B}{\partial M} \tag{6.5}$$

Using matrix calculus, $\partial (B^T B)^{-1}/\partial B$ can be derived as:

$$\frac{\partial (B^T B)^{-1}}{\partial B} = -\frac{(((B^T B)^{-1})^T \otimes I_n}{I_n \otimes (B^T B))}\frac{\partial (B^T B)}{\partial B} \tag{6.6}$$

where

$$\frac{\partial (B^T B)}{\partial B} = (B^T \otimes I_n)\frac{\partial B^T}{\partial B} + (I_n \otimes B^T)\frac{\partial B}{\partial B} \tag{6.7}$$

From the collocation matrix equation

$$B = \begin{bmatrix} \beta_{0,k}(t_1) & \cdots & \beta_{n,k}(t_1) \\ \vdots & \ddots & \vdots \\ \beta_{0,k}(t_m) & \cdots & \beta_{n,k}(t_m) \end{bmatrix}, \tag{6.8}$$

the collocation matrix B and the raw sensor measurement M is not related. However, each element of the collocation matrix is a function of time parameter t and the time parameter t is derived from chord length using raw sensor measurements. Therefore $\partial B/\partial M$ becomes:

$$\frac{\partial B}{\partial M} = \frac{\partial B}{\partial r_v}\frac{\partial r_v}{\partial M} \tag{6.9}$$

From the basis function derivative equation

$$\frac{\partial \beta_{i,k}(t)}{\partial t} = \frac{k-1}{\varepsilon_{i+k-1} - \varepsilon_i}\beta_{i,k-1}(t) - \frac{k-1}{\varepsilon_{i+k} - \varepsilon_{i+1}}\beta_{i+1,k-1}(t), \tag{6.10}$$

partial derivative of collocation matrix B with respect to time parameter sequence r_v can be derived as

$$\frac{\partial B}{\partial r_v} = \begin{bmatrix} \frac{\partial B_{1,1}}{\partial t_1} & \cdots & \frac{\partial B_{1,1}}{\partial t_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial B_{m,n}}{\partial t_1} & \cdots & \frac{\partial B_{m,n}}{\partial t_m} \end{bmatrix}. \tag{6.11}$$

And $\partial r_v / \partial M$ can be derived using chain rule:

$$\frac{\partial r_v}{\partial M} = \frac{\partial r_v}{\partial E}\frac{\partial E}{\partial C}\frac{\partial C}{\partial M} \tag{6.12}$$

with

$$\begin{aligned} E &= [\triangle e_1, ..., \triangle e_{m-1}] \\ C &= [\triangle c_1, ..., \triangle c_{m-1}] \end{aligned} \tag{6.13}$$

where $\triangle e_j$ is the real length of the curve piece j and

$$\triangle c_j = ||d_{j+1} - d_j||$$

is the chord length of the curve piece j.

From (6.1), the Jacobian of c_j w.r.t E can be derived as:

$$\begin{aligned} \frac{\partial t_j}{\partial E} &= \partial(l_j / l_{m-1})/\partial E \\ &= (l_{m-1}(\partial l_j/\partial E) - l_j(\partial l_t/\partial E))/l_{m-1}^2 \end{aligned} \tag{6.14}$$

where

$$\frac{\partial l_j}{\partial E} = [\frac{\partial l_j}{\partial \triangle e_1}, \ldots, \frac{\partial l_j}{\partial \triangle e_{m-1}}] \tag{6.15}$$

As the function of the curve is unknown, it is not possible to estimate the real curve piece length. But it is possible to get a bound between the chord length and the real length from [22]:

$$0 \leq \triangle e_j - \triangle c_j \leq \frac{1}{12}\triangle {e_j}^3 \|S(t)''\|^2_{[t_j, t_{j+1}]} \tag{6.16}$$

where $S(t)$ is the actual spline equation. Since the approximated spline $\hat{S}(t)$ and the real spline $S(t)$ should be close enough, we have:

$$\triangle e_j \leq \triangle c_j + \frac{1}{12}\triangle {c_j}^3 \|\hat{S}(t)''\|^2_{[t_j, t_{j+1}]} \tag{6.17}$$

$\|\hat{S}(t)''\|_{[t_j,t_{j+1}]}$ can be derived using (6). Therefore $\frac{\partial E}{\partial C}$ and $\frac{\partial C}{\partial M}$ can be calculated.

6.3 Data Association

In point based SLAM problem, data association process is to associate the observations to the point features in the state. However, for the proposed curve based SLAM method, features are arbitrary B-Splines. Identifying corresponding pair is insufficient. Correlation between the new observation and spline estimate $\hat{S}$ need to be identified.

Using laser range finder as an example, this section discusses the data association process for the proposed B-Spline based SLAM. It contains two stages: 1). Pairing observation with B-Splines in the map. 2). Identifying time parameter sequence r_v for new observation.

6.3.1 Pairing observation with state objects

Suppose at step n, a new scan B_n has been acquired. In order to perform the pairing, the new scan and the map estimate $V_{n,u}$, which contains u splines, need to be in the same coordinate system.

Similar with the data association process for point feature based SLAM, current scan B_n and previous scan B_{n-1} has been first transferred to the same coordinate system using odometry information (when the odometry contain large uncertainties, scan matching algorithm can be applied to get more accurate relative pose information).

After B_n and $V_{n,u}$ being in the same coordinate system, the following process were performed: *a*). Scan B_n has been segmented result in v segments corresponding to different spline objects. *b*). For segment $i(1 \leq i \leq v)$ the laser beam angle of extreme points ($Seg_{i,s}$ and $Seg_{i,e}$) are calculated. These angles are compared against the laser beam angle of extreme points ($\hat{S}_{j,s}$ and $\hat{S}_{j,e}$) for map spline $j(1 \leq j \leq u)$. Also, taking into account of range information for segment i, the pairing process completes. Fig. 6.1. illustrates this idea.

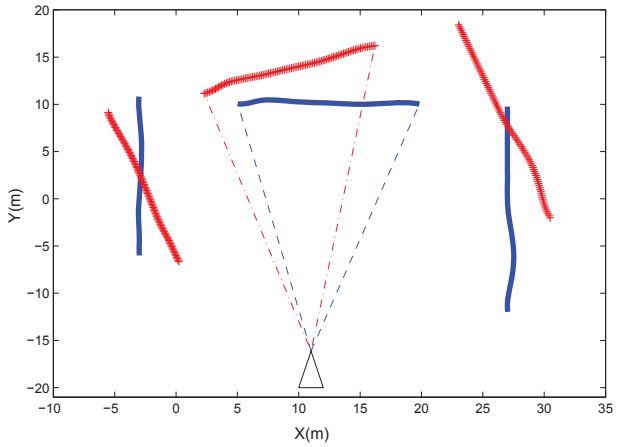


Figure 6.1: Map estimates are shown in blue line, new scan points are shown in red crosses. Laser beam angle for extreme points of data from segment i and map spline j are calculated. If $\hat{S}_{j,s} \leq Seg_{i,s} \leq \hat{S}_{j,e}$ or $\hat{S}_{j,s} \leq Seg_{i,e} \leq \hat{S}_{j,e}$ then segment i is associated with B-Spline j

6.3.2 Identify time sequence for new observations

For the proposed observation model, features are arbitrary curves represented using B-Spline. We begin with initializing a set of B-Splines derived from appropriately segmented scan data and include these splines in the state vector. The term "full spline" is used to describe the B-Splines represented in the state vector. As new laser scans been acquired, new observation about "full splines" become available. The new observation may be about the whole "full spline", part of the "full spline" or previously un-observed part of the "full-spline". In the following we use "part of spline" to describe part of the "full spline" and "extension of spline" to describe previously un-observed part of the "full-spline". Whenever "extension of spline" is observed, the "full-spline" will be updated.

To be able to estimate the same set of control points of a spline for various observation, the time parameter sequence r_v need to be uniquely defined which is invariant to the observation points. Thus the time parameter sequence for the new spline data is calculated according to the current "full spline".

In order to compute the time parameter sequence for new observation data, only the start point and end point of the observed spline data need to be corresponded with the

spline estimate $\hat{S}$. The ray tracing [77] method has been used to find the time parameter t for the extreme points. Suppose time parameter for start and end point has been identified as t_s and t_e, time parameter for remaining spline data points can be calculated by a modified version of (6.1):

$$\begin{aligned} l_1 &= 0 \\ l_j &= \sum_{i=1}^{j-1} \|d_{i+1} - d_i\| \\ l_{m-1} &= \sum_{i=1}^{m-1} \|d_{i+1} - d_i\| \\ t_j &= l_j(t_e - t_s)/l_{m-1} + t_s \end{aligned} \tag{6.18}$$

6.3.3 Spline fitting for new observations

After the relation between new observation and current "full spline" $\hat{S}_j$ has been identified and the time parameter sequence of the new spline data has been computed, the observation model of the new spline data can be derived.

Case 1: re-observe "full spline"

If the new observation is made to whole $\hat{S}$, control points and the associated covariance matrix for current observation can be derived using the process detailed in Section 6.2.

Case 2: re-observe "part of the spline"

When the new observation corresponds to part of the $\hat{S}$, we can not simply use the spline fitting process detailed in Section 6.2 to estimate the control point.

Assume the knot vector for an order k spline is (2.52) time parameter sequence for the new observation is $r_v = [t_1, ..., t_m]$ while $0 \leq \varepsilon_a < t_1 < t_m < \varepsilon_b \leq 1$. From the collocation matrix equation (6.8), the collocation matrix for the new observation can be derived as

$$B_{new} = \begin{bmatrix} \beta_{a,k}(t_1) & \cdots & \beta_{b,k}(t_1) \\ \vdots & \ddots & \vdots \\ \beta_{a,k}(t_m) & \cdots & \beta_{b,k}(t_m) \end{bmatrix}. \tag{6.19}$$

Using our proposed method in Section 6.2.2, control points X_a to X_b and associated covariance matrix can be derived.

Case 3: observe "extension of the spline"

If unexplored part of a spline has been observed, $\hat{S}$ needs to be updated. Common area between $\hat{S}$ and the new observation need to be firstly identified (Fig. 6.2. illustrates this). An updated length of the spline can be calculated. Time parameter sequence for the new observation according to new length will be calculated. Estimate of the control points can be derived from method described in *Case* 2.

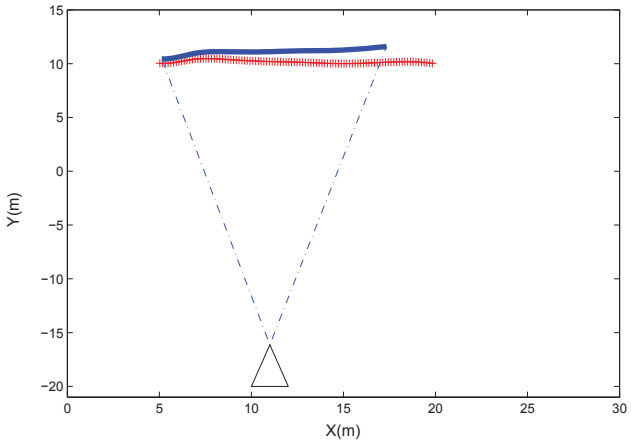


Figure 6.2: Map estimate are shown in blue line, new scan points are shown in red crosses. Laser beam angle for extreme points of map spline $\hat{S}$ are calculated. After common area being identified, the spline length has been updated.

6.4 Consistency of the Observation Model

To evaluate the consistency of the proposed observation model, simulations have been carried out. In our simulation, the range finder observations are generated by finding the intersection points between the reference spline and the artificial laser beams from a fixed robot pose. The field view of the sensor is $[-\frac{\pi}{2}, \frac{\pi}{2}]$ and the sensor range used is $6m$.

6.4.1 Test method

The consistency of the proposed observation model is examined in simulation by performing Monte Carlo runs. For example, to perform a large number of simulations with independent sensor noises and use the resulting state estimates to compute the average normalized estimation error squared (NEES) of different runs and then perform a chi-square test [32]. The NEES of each run is given by

$$NEES = (\hat{x} - x_{true})^T P^{-1} (\hat{x} - x_{true}) \tag{6.20}$$

where $\hat{x}$ is the state estimate and P is the covariance matrix and x_{true} is the ground truth of the state.

Because the ground truth of the control points is needed to perform the consistency analysis of the map, we first discuss how to get these ground truth control points in simulation.

6.4.2 Estimate the "ground truth" of control points using noise free simulation data

It could be noted that, the ground truth of the control points are not trivial to be defined prior due to the time sequence given in Fig. 2.3 which is evenly distributed, is *NOT* the same as the time parameter derived from the curve length.

Here we propose to use a good approximation of the control points as the "ground truth". In order to estimate the control points, the curve length of the reference spline must be approximated as accurate as possible. Here a small angular resolution (0.1^o) was used and no noise is added in the simulated scan data. The time sequence was defined using equation (6.18). Using the spline fitting method described in Section 2.7.3, an estimate of the control points is obtained. The associated covariance matrix computed from the method described in Section 6.2 is very small. To validate this, a new spline equation was derived using the estimated "ground truth" control points. The curve of the new spline was compared with the reference spline and the difference between the two curves is undistinguishable.

To further prove the validity of the "ground truth" control points using the "Affine

Invariance Property" defined in Section 2.7.3, new noise-free observation of the spline was taken from another position. The control points estimated from the second scan were derived and transferred back to the coordinate defined by the first pose. The result is shown in Fig. 6.3. It can be observed that, the control points estimates overlap nicely with the "ground truth control points".

6.4.3 Estimate the control points through noisy scan data

To simulate the real laser data, we use 0.5^o for the laser resolution and we used a zero mean Gaussian noise with $6mm$ standard deviation to add on the range readings. The control points estimated from the proposed spline fitting algorithm in Section 6.2 and the associated 2σ uncertainty ellipses are plotted and compared with the "ground truth" control points in Fig. 6.4.

To investigate the consistency of the observation model, 50 simulation data sets are generated and the average NEES obtained is 33.45, while the chi-square distribution for the 95% confidence level is 33.92.

6.4.4 Estimate control points when "part of spline" is observed

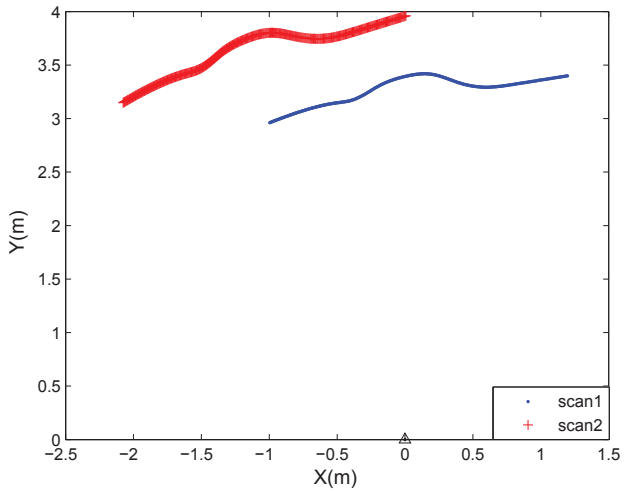
A simulation result for the scenario when only "part of spline" is observed is shown in Fig. 6.5(a). The uncertainty ellipses are shown in Fig. 6.5(b).

Again, 50 simulation data sets with added independent noise (from the second scan only) are generated. The average NEES obtained is 16.03, while the chi-square distribution for the 95% confidence level is 26.29.

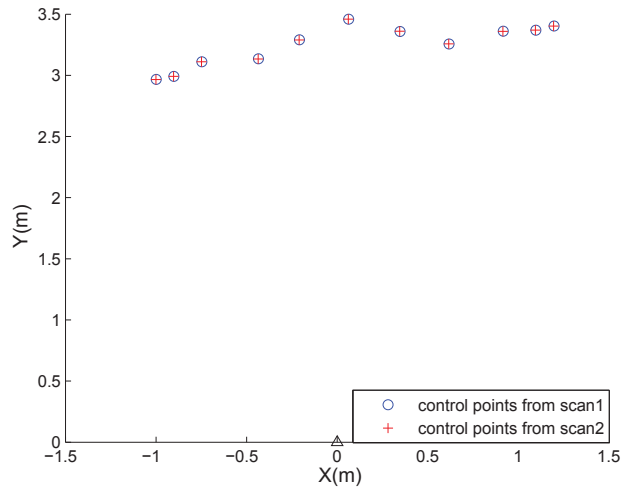
6.5 SLAM Results

6.5.1 SLAM result using simulation data

A simulation experiment containing 9 splines in the environment and 46 robot poses was created to further evaluate the consistency of the proposed SLAM algorithm. The simulation environment is shown in Fig. 6.6.



(a) Two noise-free scans from different poses



(b) "Ground truth" control points from the two noise-free scans (the control points obtained from the second scan is transferred to the coordinate system defined by the first pose)

Figure 6.3: Estimate "ground truth" control points

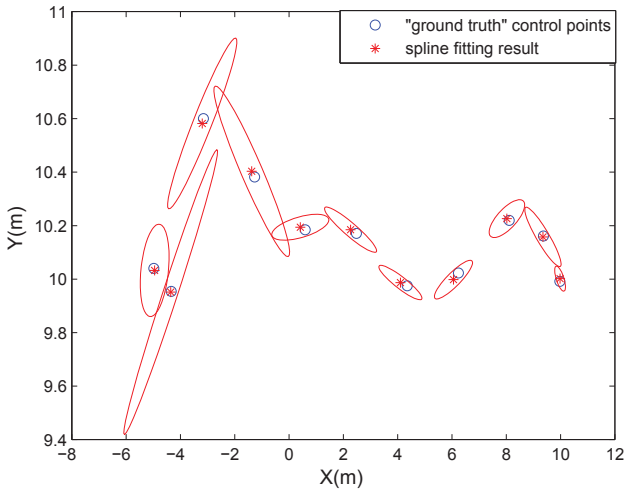


Figure 6.4: Spline fitting with noisy data

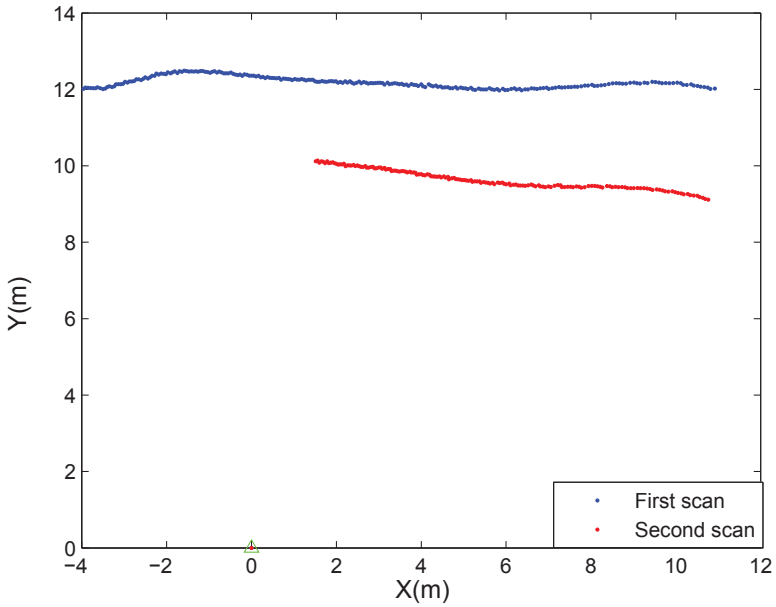
The SLAM results obtained using I-SLSJF [35] is shown in Fig. 6.7(a). Comparing with the "ground truth", the estimate of control points and robot poses appear to be very accurate. To evaluate the consistency of the observation model and the SLAM results, the NEES comparison is plotted Fig. 6.7(b). The NEES of the final estimate (for the final map containing 99 control points and 46 poses) obtained is 391.16, while the the 99% probability concentration region of the chi-square distribution is [0, 399.32].

We also test the proposed observation model using the SDR based approach. The comparison between the result from LS approach and SDR approach is shown in Fig 6.8.

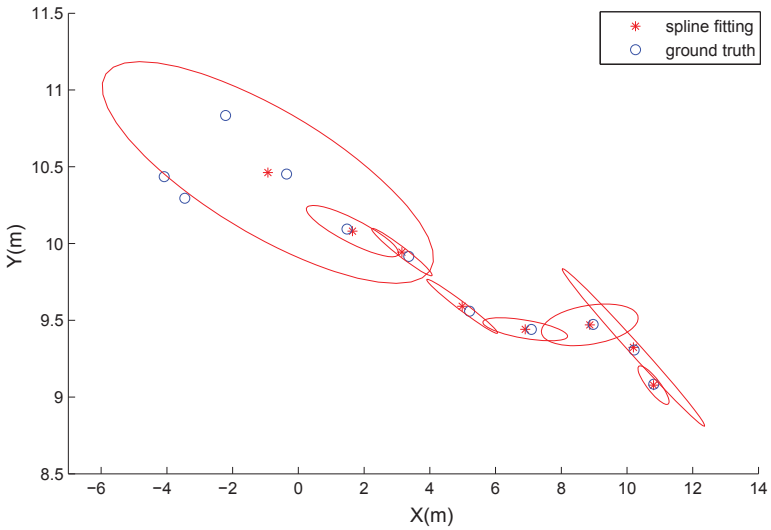
6.5.2 SLAM using real data

Using the 47 scans of Intel dataset from the Robotics Data Set Repository [38] described in Section 5.6.4 we further validate the proposed algorithm.

Due to the imperfection of the data association, manual association are performed on some of the scans. In the experiment, cubic splines with 11 knots were used. Fig. 6.9(b) shows the control points estimates obtained from LS. The map contains 29 cubic splines. Each spline contains 7 control points. Fig. 6.10 depicts the map using cubic splines



(a) Raw data for observing part of a spline. First scan is "full spline", second scan is "part of spline"



(b) Spline fitting result when "part of spline" is observed. The 3 left most control points are not estimated here, as this range has not been observed

Figure 6.5: Spline fitting with noisy data when "part of spline" is observed

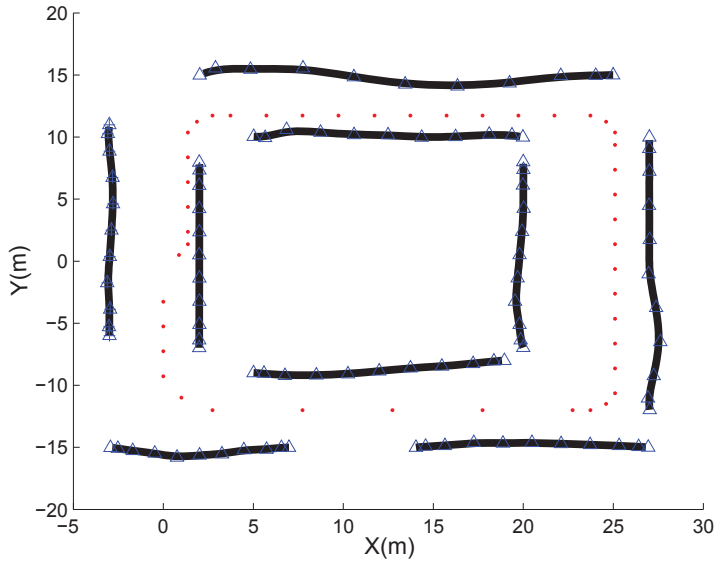


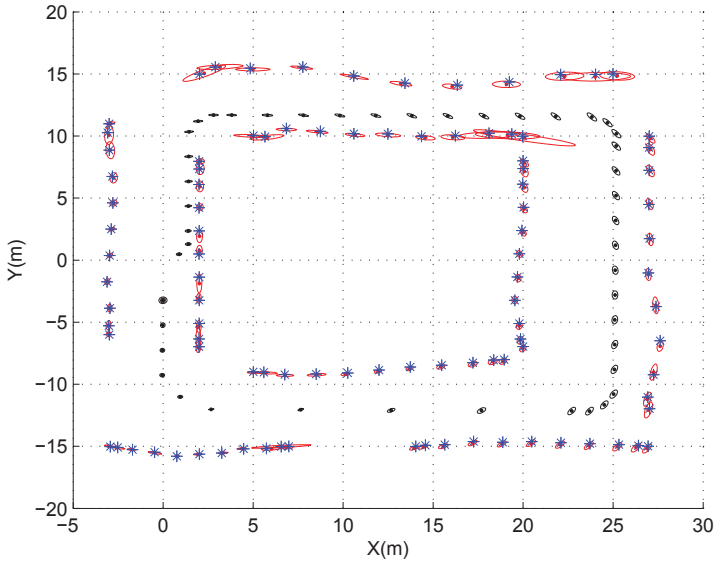
Figure 6.6: A simulation environment with 9 splines, red dots show the robot poses, blue triangles show the "ground truth" of the control points

derived from the control point estimates.

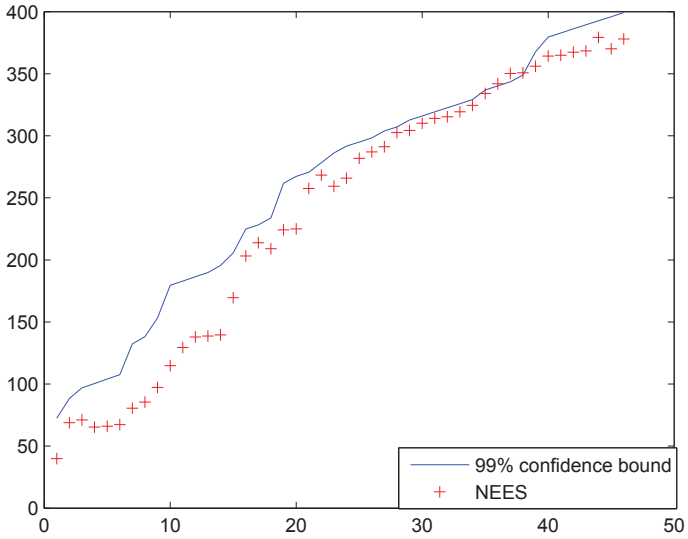
The comparison between the result from I-SLSJF approach and SDR approach for this case is shown in Fig 6.11. As the proposed SDR approach is a convex optimization based method, it only has one global minimum. Therefore, it does not provide any uncertainty information. However, it can be observed from Fig 6.11 that, results acquired from the SDR approach is very close to the mean of I-SLSJF result.

6.6 Summary

In this chapter, a new observation model for B-Spline based SLAM is proposed. Method to estimate the covariance matrix for control points in a B-Spline based SLAM algorithm is also proposed. The errors in the control points estimates originates from two dependent sources: the raw data error and the chord length estimate error. It has been shown that, under various conditions, the proposed error analysis leads to consistent spline fitting result.



(a) SLAM result using the simulation data. Blue star show the "ground truth" control points. Estimated control points almost coincide with "ground truth"



(b) Comparison of NEES v.s. 99% confidence gate at each step

Figure 6.7: Simulation result

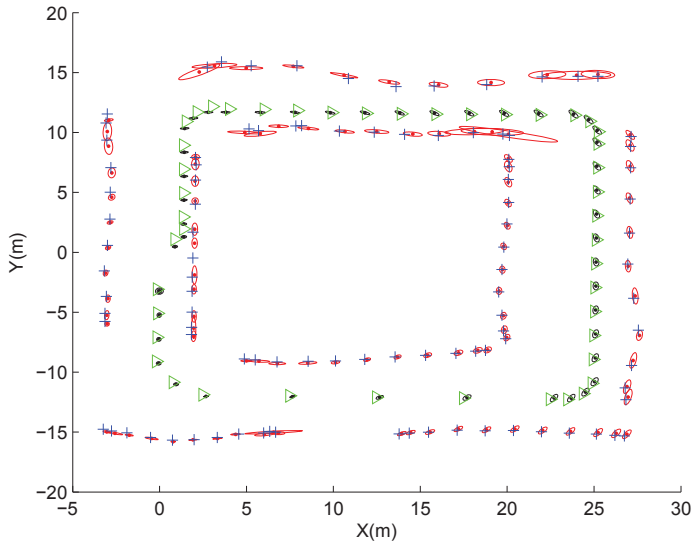
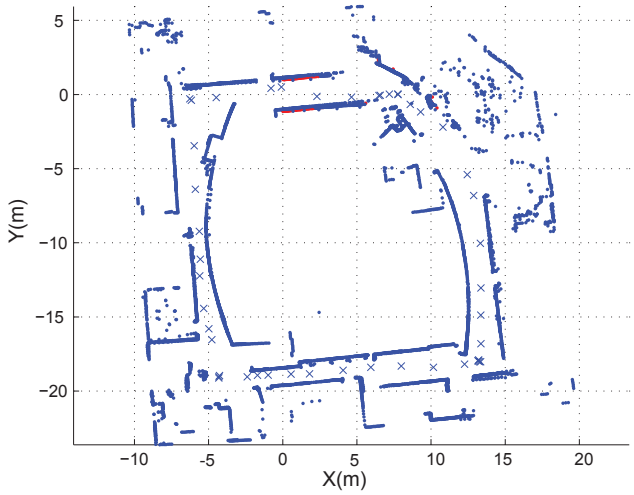
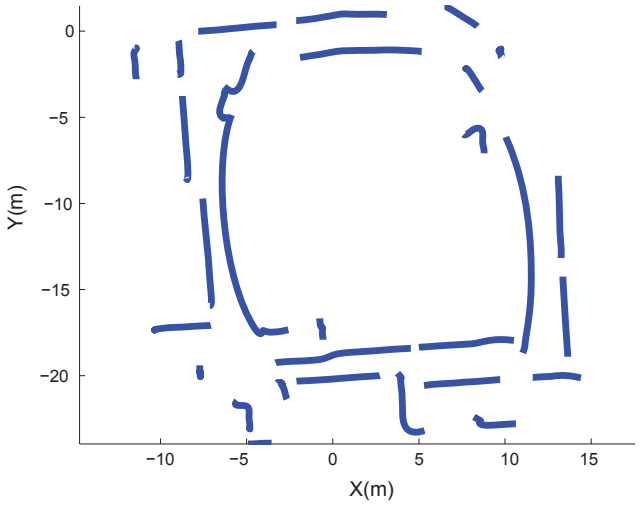


Figure 6.8: Comparison of LS result with the SDR result for the simulation data (black dot is the LS robot position estimate; red dot is the LS feature position estimate; green triangle is the SDR robot position estimate; blue cross is the SDR feature position estimate).



(a) Result after applying least squares on ICP result



(b) Spline map derived by applying least squares on the new observation model

Figure 6.9: SLAM result using the Intel dataset

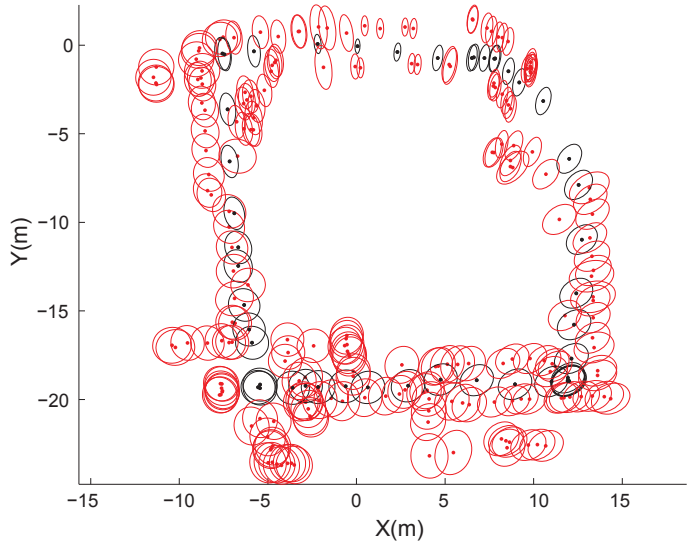


Figure 6.10: The location of the estimated control points

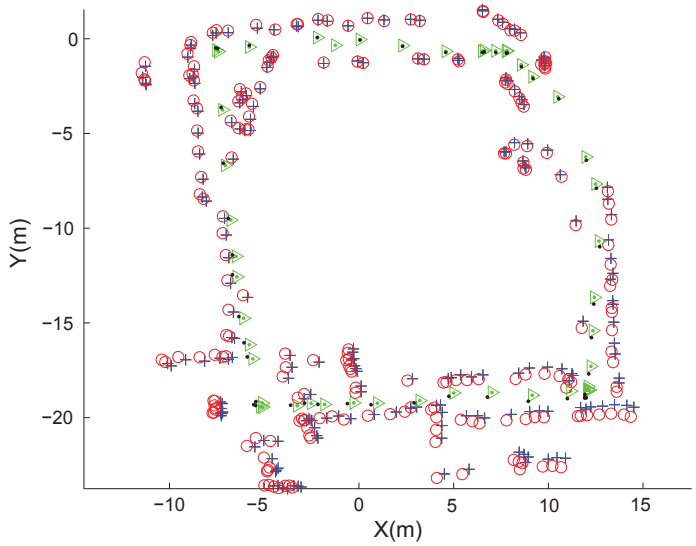


Figure 6.11: Comparison of I-SLSJF result with the SDR result for the Intel data (black dot is the I-SLSJF robot position estimate; red dot is the I-SLSJF feature position estimate; green triangle is the SDR robot position estimate; blue cross is the SDR feature position estimate).

Using the new observation model, the curve feature based SLAM can be transferred into point feature based SLAM. Some SLAM results using simulation and real data further proves that the estimate is consistent. It has also been shown that the proposed convex optimization based approach can also be applied to the transformed curve feature SLAM problem.

7 Conclusions and Future Work

The main objective of this thesis is attempting to provide a solution to an important problem of autonomous mobile robots namely simultaneous localization and mapping (SLAM). In particular, three problems were considered: 1) The convergence of point feature based SLAM 2) Cure feature based SLAM 3) The relationship between point feature based SLAM and curve feature based SLAM. This chapter aims to summarize the principle contributions of this thesis and provides a set of future directions for extending this work.

This chapter is organized as follows. Section 7.1 highlights the major contributions. Section 7.2 suggests directions of future work in this field of research.

7.1 Summary of Contributions

The following lists the major contributions made in this thesis.

7.1.1 Convex optimization based approach for point feature SLAM

The first contribution of this thesis is the reformulation of the point feature SLAM problem. In Chapter 3, we propose to reformulate the point feature SLAM problem as a quadratically constrained optimization programming (QCQP) problem. This is achieved by introducing a new representation for robot orientation with quadric constraint and approximations on the structure of the uncertainty of measurements. The main advantage of this formulation over the least squares SLAM formulation is that, the objective function does not contain product of the state elements and becomes exactly quadratic. However, unlike the least squares SLAM formulation which does not involve any constraints, the new formulation involves a nonconvex quadratic constraint.

To overcome the nonconvexity, the semi-definite relaxation (SDR) technique has been applied to the new formulation to obtain a semi-definite programming (SDP) problem. Since SDP problems are convex problems, unique solution can be computed even when a suitable initial guess is not available. It is shown in Chapter 4 using simulation and experimental results that, the candidate solution is close to the true solution to the SLAM problem.

7.1.2 General curve feature SLAM formulation

In Chapter 5, a least squares formulation for general feature SLAM is proposed. In this formulation, the environment is represented by a set of continuous curves such as lines or B-Splines and negative information which indicates the absence of an expected sensor reading is exploited as useful information.

It has been observed that this formulation contains discontinuities, the problem becomes hard to solve. Although there are optimization based approaches such as Genetic Algorithm (GA) which are able to handle discontinuities, the computational cost involved is too high.

7.1.3 Curve feature SLAM to point feature SLAM conversion

Chapter 6 proposed a new observation model for B-Spline SLAM. The observation model used is a function of relative positions between the control points of the observed spline and the observation point. The uncertainty of the control points has been derived analytically from two sources: spline fitting error and the chord length error. It has been considered that the spline fitting error and the chord length error are not independent.

Using this new observation model, the conversion from curve feature SLAM to point feature SLAM becomes possible. Because the curve feature SLAM formulation is hard to solve, the curve feature is converted to point feature. Using the conversion method, the curve feature SLAM problem can be transferred to point feature SLAM problem and can be solved by the convex optimization based approach.

7.2 Future Work

The research documented herein provided insight into possible future directions as well as unforseen challenges. These are put forward in the following brief discussion.

7.2.1 Approximation ratio for the SDR based approach for SLAM problem

The theoretical relationship between the SDR result and the optimal SLAM solution has been derived only for the case when the sensor measurements do not contain noises. It is important to derive the worst-case approximation bounds when measurements contain noise, especially when the covariance matrices of the measurement noise are far from spherical.

7.2.2 Reducing the computational complexity of the SDR approach

Generally speaking, using interior-point algorithm, the SDP problem (3.45) can be solved with a worst case complexity of

$$O(N^{4.5}log(1/\varepsilon))$$

where $\varepsilon > 0$ is the solution accuracy. However, the computational complexity for some of the modern SLAM algorithms is

$$O(N^2)$$

To apply the proposed SDR approach, the computational complexity for the proposed approach need to be reduced.

One can even exploit the structure of the SLAM problem to build fast customized interior-point algorithm.

7.2.3 Extend the SDR based approach to other form of SLAM problem

Only the 2D, feature-based, range and bearing SLAM problem was analyzed. It is not clear whether the extension to 3D, or other SLAM problems such as range-only and bearing-only SLAM are feasible.

Bibliography

[1] M. Agrawal and K. Konolige, "FrameSLAM: From bundle adjustment to real-time visual mapping," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1066-1077, 2008.

[2] A. Aspremont and S. Boyd, "Relaxations and randomized methods for nonconvex QCQPs," *Stanford Univ., Stanford*, CA, 2003.

[3] N. Ayache and O. Faugeras. "Maintaining a representation of the environment of a mobile robot," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 6, pp. 804-819, 1989.

[4] T. Bailey and H. Durrant-Whyte, "Simultanouse localization and mapping (SLAM): Part II," *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 108-117, 2006.

[5] T. Bailey, J. Nieto, J. Guivant, M. Stevens and E. Nebot. "Consistency of the EKF-SLAM algorithm," in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, pp. 3562-3568, 2006.

[6] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, 2001. *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. John Wiley & Sons.

[7] P. Besl and N. McKay. "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239-256, 1992.

[8] P. Biswas, T. C. Liang, K. C. Toh, Y. Ye, and T. C. Wang, "Semidefinite programming approaches for sensor network localization with noisy distance measurements," *IEEE Transactions on Automatation Science and Engineering*, vol. 3, pp. 360-371, 2006.

[9] S. J. Benson and Y. Ye, "DSDP5 user guide-software for semidefinite programming," Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, Tech. Rep. ANL/MCS-TM- 227, Sep. 2005. Available online: http://www.mcs.anl.gov/ benson/dsdp

[10] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in *Proceedings of International Symposium Information Processing in Sensor Networks. ACM Press*, pp. 46-54, 2004.

[11] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. "An Atlas framework for scalable mapping," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, pp. 1899-1906, 2003.

[12] W. Burgard, C. Stachniss, G. Grisetti, B.Steder, R. Kummerle, C. Dornhege, M. Ruhnke, A.Kleiner, and J. D. Tardos. "A comparison of SLAM algorithms based on a graph of relations," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009.

[13] L. Carlone, R. Aragues, J. Castellanos and B. Bona. "A linear approximation for graph-based simultaneous localization and mapping," *Robotics: Science and Systems (RSS)*, 2011.

[14] J. A. Castellanos and J. D. Tardós, "Mobile robot localization and map building: a Multisensor fusion approach," *Boston, MA: Kluwer*, 1999.

[15] A. Censi, "An ICP variant using a point-to-line metric," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 19-25, 2008.

[16] R. Chatila and J.P. Laumond. "Position referencing and consistent world modeling for mobile robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 138-145, 1985.

[17] M. Cummins and P. M. Newman. "Highly scalable appearance-only SLAM-FABMAP 2.0," in *Robotics Science and Systems (RSS)*, 2009.

[18] F. Dellaert, J. Carlson, V. Ila, K. Ni, and Charles E. Thorpe, "Subgraph-preconditioned conjugate gradients for large scale SLAM," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2066-2571, 2010.

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

[20] G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A so-

lution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 229-241, 2001.

[21] C. Estrada, J. Neira, and J.D. Tardos. "Hierarchical SLAM: realtime accurate mapping of large environments". *IEEE Transactions on Robotics*, vol. 21, no. 4, pp. 588-596, 2005.

[22] M. S. Floater. "Arc length estimation and the convergence of polynomial curve interpolation," *BIT Num. Math*, vol. 45, 679-694, 2005.

[23] U. Frese, "A discussion of simultaneous localization and mapping," *Autonomous Robots*, vol. 20, no. 1, pp. 25-42, 2006.

[24] U. Frese. "Treemap: An O(log n) algorithm for indoor simultaneous localization and mapping," *Autonomous Robots*, vol. 21, no. 2, pp. 103-122, 2006.

[25] A. Garulli, A. Giannitrapani, A. Rossi, and A. Vicino, "Mobile robot SLAM for line-based environment representation," *IEEE Conference on Decision and Control (CDC)*, pp. 2041-2046, 2005.

[26] G. Grisetti, C. Stachniss and W. Burgard, "Improved techniques for grid mapping with Rao-Blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34-46, 2007.

[27] G. Grisetti, D. Rizzini, C. Stachniss, E. Olson, and W. Burgard, "Online constraint network optimization for efficient maximum likelihood map learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1880-1885, 2008.

[28] G. Grisetti, D. L. Rizzini, C. Stachniss, E. Olson and W. Burgard. "Online constraint network optimization for efficient maximum likelihood mapping," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, pp. 1880-1885, 2008.

[29] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard; "A tree parameterization for efficiently computing maximum likelihood maps using gradient decent," *Robotics Science and Systems (RSS)*, 2007.

[30] J. E. Guivant and E. M. Nebot, "Optimization of the simultaneous localization and map building (SLAM) algorithm for real time implementation," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 242-257, 2001.

[31] J. Hoffmann, M. Spranger, D. Goehring, and M. Juengel. "Making use of what

you don't see: Negative information in markov localization," in *Proceedings of the 2005 IEEE/RSJ International Conference of Intelligent Robots and Systems (IROS)*, pp.2947-2952, 2005.

[32] S. Huang and G. Dissanayake. "Convergence and consistency analysis for Extended Kalman Filter based SLAM," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1036-1049, 2007.

[33] S. Huang, Y. Lai, U. Frese, and G. Dissanayake, "How far is SLAM from a linear least squares problem," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3011-3016, 2010.

[34] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis. "Analysis and improvement of the consistency of Extended Kalman Filter-based SLAM," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 473-479, 2008.

[35] S. Huang, Z. Wang, G. Dissanayake, and U. Frese, "Iterated SLSJF: A sparse local submap joining algorithm with improved consistency," in *Proceedings of Australasian Conference on Robotics and Automation (ACRA),* 2008. Available online: http://www.araa.asn.au/acra/acra2008/papers/pap102s1.pdf

[36] S. Huang, Z. Wang, G. Dissanayake and U. Frese. "Iterated D-SLAM map joining: Evaluating its performance in terms of consistency, accuracy and efficiency," *Autonomous Robots*, vol. 27, pp. 409-429, 2009.

[37] J. Hoschek, and D. Lasser. *Fundamentals of Computer Aided Geometric Design*, AK Peters 1993.

[38] A. Howard and N. Roy. The robotics data set repository (Radish). Available online: http://radish.sourceforge.net

[39] B. Jensen and R. Siegwart, "Scan alignment with probabilistic distance metric," in *Proceedings of the IEEE/RSJ International conference on Intelligent Robots and Systems (IROS)*, pp. 2191-2196, 2004.

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

[41] W. Koch. Utilizing, "Negative Information to Track Ground Vehicles Through Move-stop-move Cycles," in *Proceedings of the SPIE*, vol. 5429, pp. 273¨C283, 2004.

[42] K. Konolige, M. Agrawal, "FrameSLAM: From bundle adjustment to real-time visual mapping," *IEEE Transactions on Robitics*, vol. 24, no. 5, pp. 1066-1077, 2008.

[43] K. Konolige, G. Grisetti, R. Kummerle, B. Limketkai, and R. Vincent, "Efficient sparse pose adjustment for 2D mapping," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 22-29, 2010.

[44] J. Kurlbaum and U. Frese, "A benchmark data set for data association." Technical Report, University of Bremen, available online: http://www.sfbtr8.uni-bremen.de/reports.htm, also available on http://radish.sourceforge.net/

[45] C. Leung, S. Huang and G. Dissanayake, "Active SLAM for structured environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1898-1903, 2008.

[46] M. Liu, S. Huang and G. Dissanayake. "A new observation model for B-Spline SLAM," in *Proceedings of Australasian Conference on Robotics and Automation (ACRA)*, 2009.

[47] M. Liu, S. Huang, G. Dissanayake and S. Kodagoda. "Towards a consistent SLAM algorithm using B-Splines to represent environments," in *Proceedings of the IEEE/RSJ International conference on Intelligent Robots and Systems (IROS)*, pp. 2065-2070, 2010.

[48] F. Lu and E. Milios. "Globally consistent range alignment for environment mapping," *Autonomous Robots*, vol. 4, pp.333-349, 1997.

[49] K. W. K. Lui, W.-K. Ma, H. C. So, and F. K. W. Chan, "Semi-definite programming algorithms for sensor network node localization with uncertainties in anchor positions and/or propagation speed," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 752-763, 2009.

[50] Z. Q. Luo, W. K. Ma, A. M. C. So, Y. Ye, and S. Z. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 20-34, 2010.

[51] R. Madhavan, G. Dissanayake, and H. F. Durrant-Whyte, "Autonomous underground navigation of an LHD using a combined ICP-EKF approach," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3703-3708, 1998.

[52] J. R. Magnus. "On the concept of matrix derivative," *Journal of Multivariate Analysis*, vol. 101, no. 9, pp. 2200-2206, 2010.

[53] J. L. Martínez, J. González, J. Morales, A. Mandow and A. J. García-Cerezo. "Mobile robot motion estimation by 2D scan matching with genetic and iterative closest point algorithms," *Journal of Field Robotics*, vol. 23, no. 1, pp. 21-34, 2006.

[54] J. Minguez, F. Lamiraux, and L. Montesano, "Metric-based scan matching algorithms for mobile robot displacement estimation," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp.1047-1054, 2006.

[55] L. Montesano, J. Minguez, and L. Montano, "Probabilistic scan matching for motion estimation in unstructured environments," in *Proceedings of the IEEE/RSJ International conference on Intelligent Robots and Systems (IROS)*, pp. 3499-3504, 2005.

[56] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proceedings of AAAI National Conference on Artificial Intelligence*, 2002.

[57] M. Montemerlo and S. Thrun. "Simultaneous Localization and Mapping with Unknown Data Association Using FastSLAM". in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1985-1991, 2003.

[58] K. Murphy. "Bayesian map learning in dynamic environments," in *Advances in Neural Information Processing Systems 11*, MIT Press, 1999.

[59] H. Nielsen. "Damping Parameter in Marquardt's Method," Technical Report IMM-REP-1999-05, Technical University of Denmark, 1999. Available at http://www.imm.dtu.dk/ hbn

[60] P. Newman, D. Cole, and K. Ho. "Outdoor SLAM using visual appearance and laser ranging," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1180-1187, 2006.

[61] K. Ni, D. Steedly, and F. Dellaert, "Tectonic SAM: Exact, out-of-core, submap-based SLAM," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1678-1685.

[62] J. Nieto, T. Bailey and E. Nebot. "Scan-SLAM: Combining EKF-SLAM and scan

correlation," in *Proceedings of the International Conference on Field and Service Robotics*, 2005.

[63] E. Olson, J. Leonard, and S. Teller, "Fast iterative optimization of pose graphs with poor initial estimates," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2262-2269, 2006.

[64] L. Pedraza, G. Dissanayake, J. Valls Miro, D. Rodriguez-Losada, and F. Matia. "Extending the limits of feature-based SLAM with B-Splines," *IEEE Transactions on Robotics*, vol. 25, pp. 353-366, 2009.

[65] L. Piegl and W. Tiller. *The NURBS book*. 2nd ed. Berlin: Springer-Verlag 1997.

[66] P. Pinies, J.D. Tardos, "Large-scale slam building conditionally independent local maps: Application to monocular vision," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1094-1106 (2008).

[67] D. L. Rizzini, "Towards a closed-form solution of constraint networks for Maximum Likelihood mapping," in *International Conference on Advanced Robotics (ICAR),* 2009.

[68] D. Rodriguez-Losada, F. Matia, A. Jimenez, and R. Galan. "Consistency improvement for SLAM-EKF for indoor environments," In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 418-423, 2006.

[69] D. Ruppert and R.J Carroll. "Trimmed least squares estimation in the linear model," *Journal of the American Statistical Association* vol. 75, pp. 828-838, 1980.

[70] S. Sarkka, T. Tamminen, A. Vehtari, and J. Lampinen, "Probabilistic Methods in Multiple Target Tracking", Research Report B36, Technical report, Laboratory of Computational Engineering Helsinki University of Technology, 2004.

[71] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Robotics Science and Systems*, 2009.

[72] R. Smith, M. Self, and P. Cheeseman. "A stochastic map for uncertain spatial relationships," in *International Symposium of Robotics Research*, pp. 467-474, 1987.

[73] R. Smith, M. Self, and P. Cheeseman. "Estimating uncertain spatial relationships in robotics," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 850, 1987.

[74] A. M. C. So and Y. Ye, "Theory of semidefinite programming for sensor network localization," *Math. Program*, Series B, vol. 109, no. 2-3, pp. 367-384, 2007.

[75] J. F. Sturm. "Using SeDuMi 1.02, a MATLAB toolbox for optimization overy symmetric cones," *Optimization Methods and Software*, 11-12: 625-653, 1999.

[76] N. Sunderhauf, K. Konolige, S. Lacroix, and P. Protzel, "Visual odometry using sparse bundle adjustment on an autonomous outdoor vehicle," in *Tagungsband Autonome Mobile Systeme*. 2005.

[77] A. J. Sweeney and R. H. Barrels. "Ray tracing free-form B-spline surfaces," *IEEE Computer Graphics and Application*, 6(2):41-49, Feb. 1986.

[78] J. Tardos. "Representing partial and uncertain sensorial information using the theory of symmetries," In *Proceeding of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1799-1804, 1992,

[79] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*, pp. 171-172. MIT Press, 2005.

[80] Z. Wang, S. Huang, and G. Dissanayake, "D-SLAM:Adecoupled solution to simultaneous localization and mapping," *International Journal of Robotics Research*, vol. 26, no. 2, pp. 187-204, 2007.

[81] S. B. Williams, G. Dissanayake, and H. Durrant-Whyte. "An efficient approach to the simultaneous localisation and mapping problem," *Proceeding of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 406-411, 2002.