Laser-Based 3D Mapping and Navigation in Planetary Worksite Environments

by

Chi Hay Tong

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Aerospace Science and Engineering
University of Toronto

Abstract

Laser-Based 3D Mapping and Navigation in Planetary Worksite Environments

Chi Hay Tong
Doctor of Philosophy
Graduate Department of Aerospace Science and Engineering
University of Toronto
2013

For robotic deployments in planetary worksite environments, map construction and navigation are essential for tasks such as base construction, scientific investigation, and in-situ resource utilization. However, operation in a planetary environment imposes sensing restrictions, as well as challenges due to the terrain. In this thesis, we develop enabling technologies for autonomous mapping and navigation by employing a panning laser rangefinder as our primary sensor on a rover platform.

The mapping task is addressed as a three-dimensional Simultaneous Localization and Mapping (3D SLAM) problem. During operation, long-range 360° scans are obtained at infrequent stops. These scans are aligned using a combination of sparse features and odometry measurements in a batch alignment framework, resulting in accurate maps of planetary worksite terrain.

For navigation, the panning laser rangefinder is configured to perform short, continuous sweeps while the rover is in motion. An appearance-based approach is taken, where laser intensity images are used to compute Visual Odometry (VO) estimates. We overcome the motion distortion issues by formulating the estimation problem in continuous time. This is facilitated by the introduction of Gaussian Process Gauss-Newton (GPGN), a novel algorithm for nonparametric, continuous-time, nonlinear, batch state estimation.

Extensive experimental validation is provided for both mapping and navigation components using data gathered at multiple planetary analogue test sites.

Dedication

To Topaz and Zoe, my love and joy.

Acknowledgements

This thesis would not have been possible without the help of many people. It is thanks to these people that I get the opportunity to play with robots every day.

First and foremost, I would like to thank my family for their love and support over these many years. To my parents, Tack and Juliana, you made this long journey possible. Thank you for letting me follow my heart, and offering support when I needed it most. To my brother, Chi-Yan, it has been great to grow up with you. Finally, to my wife, Topaz, and my daughter, Zoe, you have always been beside me every step of the way. We make a great team.

I would like to thank my supervisor, Prof. Tim Barfoot, for giving me my start in this field. Your expertise in mathematics revealed to me the beauty of theory, and your honest drive for research excellence has shown me what it takes to succeed in academia. May tenured life lead to greater and grander things.

To my colleagues at the Autonomous Space Robotics Lab (ASRL), it was a great pleasure working with all of you. Specifically, I would like to thank Paul Furgale for collaborating on SUSURF, donating many whiteboard hours, and showing me that growing up is optional. I would also like to thank Braden Stenning for his work on the field rovers, Rehman Merali for his participation in the 3D Site Map project, Goran Basic for his great stories and hardware ability, Keith Leung for his technical advice, and Jon Gammell for lending a helpful ear. Since space limitations prevent me from personalizing all of my messages, to Pat, Hien, Colin, Andrew, Hang, Billy, Konstantine, Trevor, Yoni, Sean, Erik, Chris, Anjani, Sara, and Peter, thank you. I hope we get the opportunity to work together once again in the future.

Contents

1 Introduction **1**

2 Discrete-Time Batch SLAM **7**

2.1 Introduction 7

2.2 Related Work 8

2.3 Discrete-Time Batch Gauss-Newton 9

2.3.1 Background 9

2.3.2 Problem Statement 10

2.3.3 Optimization 11

2.3.4 Summary 13

3 Continuous-Time Batch SLAM **14**

3.1 Introduction 14

3.2 Related Work 17

3.3 Gaussian Process Gauss-Newton 19

3.3.1 Background 19

3.3.2 Problem Statement 20

3.3.3 Weight-Space Derivation 21

3.3.4 Function-Space Derivation 26

3.3.5 Linear Prediction Equation 27

3.3.6 Measurement Interpolation 29

3.3.7 Summary 30

3.4 Experimental Validation 30

3.4.1 Overview 31

3.4.2 GPGN Implementation 31

3.4.3 Discrete-Time Batch Gauss-Newton Implementation 33

3.4.4 Simulation Results 33
3.4.5 Hardware Experiment 35
3.5 Conclusion 39

4 State Estimation in Three Dimensions 41
4.1 Introduction 41
4.2 State Parametrization 42
4.2.1 Useful Properties 43
4.3 Perturbations and Uncertainty 45
4.4 Measurement Model Linearization 46
4.4.1 Camera Model 46
4.4.2 Odometry 47
4.5 Summary 48

5 Planetary Worksite Mapping 49
5.1 Introduction 49
5.2 Related Work 53
5.3 Architecture Overview 56
5.4 Measurements 57
5.4.1 Natural Feature Extraction and Matching 57
5.4.2 Odometry 60
5.5 Batch Alignment 61
5.5.1 System Models 61
5.5.2 Optimization 63
5.5.3 Estimate Reference Frame Selection 64
5.5.4 Final Map Construction 65
5.6 Heterogeneous Measurement Outlier Rejection 65
5.7 Experiment Metrics 68
5.7.1 Localization Accuracy 68
5.7.2 Map Quality 70
5.8 UTIAS Indoor Rover Test Facility 70
5.8.1 Overview 70
5.8.2 Results 73
5.9 CSA Mars Emulation Terrain 76

5.9.1 Overview . . . 76
5.9.2 Results . . . 78
5.10 Discussion . . . 81
5.10.1 Failure Modes . . . 81
5.10.2 Possible Improvements . . . 83
5.11 Conclusion . . . 84

6 Laser-Based Visual Odometry 86
6.1 Introduction . . . 86
6.2 Related Work . . . 88
6.3 Architecture Overview . . . 90
6.4 Feature Measurements . . . 91
6.5 Continuous-Time State Estimation . . . 93
6.5.1 System Models . . . 94
6.5.2 Optimization . . . 96
6.5.3 Sliding Window . . . 96
6.6 Experimental Validation . . . 98
6.6.1 Outdoor Traverse . . . 98
6.6.2 UTIAS Indoor Rover Test Facility . . . 101
6.7 Discussion . . . 105
6.7.1 Failure Modes . . . 105
6.7.2 Possible Improvements . . . 106
6.8 Conclusion . . . 107

7 Conclusion 108

A Physically Motivated Gaussian Processes 112
A.1 Gaussian Process for a Moving Point . . . 112
A.1.1 General Derivation . . . 112
A.1.2 Specific Example of a White Noise Prior on $\ddot{\mathbf{x}}(t)$ . . . 115
A.2 Incorporating Derivative-Type Measurements . . . 118

B Normalized Innovation Squared Test 120
B.1 Measurement Models . . . 120
B.2 Hypothesis Model . . . 121

B.3 Statistical Test 121

Bibliography **124**

Notation

a : Symbols in this font are real scalars.

$\mathbf{a}$: Symbols in this font are real column vectors.

$\mathbf{A}$: Symbols in this font are real matrices.

$\boldsymbol{a}$: Symbols in this font are vectors expressed in homogeneous coordinates.

$\sim \mathcal{N}(\mathbf{a}, \mathbf{B})$: Gaussian distributed with mean vector $\mathbf{a}$ and covariance matrix $\mathbf{B}$.

$\sim \mathcal{GP}(\boldsymbol{\alpha}(t), \boldsymbol{\mathcal{B}}(t, t'))$: Gaussian process with mean function $\boldsymbol{\alpha}(t)$ and covariance function $\boldsymbol{\mathcal{B}}(t, t')$.

$E[\cdot]$: The expectation operator.

$(\cdot)^{\times}$: The cross-product operator that produces a 3 × 3 skew-symmetric matrix from a 3 × 1 column vector.

$(\cdot)^{\boxplus}$: A matrix operator that produces a 4 × 4 matrix from a 6 × 1 vector.

$(\cdot)^{\boxminus}$: A matrix operator that produces a 4 × 6 matrix from a 4 × 1 vector.

$(\cdot)^{\boxdot}$: A matrix operator that produces a 6 × 6 matrix from a 4 × 4 matrix.

$\mathbf{1}$: The identity matrix.

$\mathbf{0}$: The zero matrix.

$\underrightarrow{\mathcal{F}}_a$: A reference frame in three dimensions.

$\mathbf{p}_a^{c,b}$: A vector from point b to point c (denoted by the superscript) and expressed in $\underrightarrow{\mathcal{F}}_a$ (denoted by the subscript).

$\boldsymbol{p}_a^{c,b}$: The vector $\mathbf{p}_a^{c,b}$ expressed in homogeneous coordinates.

$\mathbf{C}_{a,b}$: The 3 × 3 rotation matrix that transforms vectors from $\underrightarrow{\mathcal{F}}_b$ to $\underrightarrow{\mathcal{F}}_a$: $\mathbf{p}_a^{c,b} = \mathbf{C}_{a,b}\mathbf{p}_b^{c,b}$.

$\boldsymbol{T}_{a,b}$: The 4 × 4 transformation matrix that transforms homogeneous points from $\underrightarrow{\mathcal{F}}_b$ to $\underrightarrow{\mathcal{F}}_a$: $\boldsymbol{p}_a^{c,a} = \boldsymbol{T}_{a,b}\boldsymbol{p}_b^{c,b}$.

Chapter 1

Introduction

With the renewed worldwide interest in developing new methods for establishing a permanent presence on extraterrestrial surfaces, it is likely that robotics will play a large role in the precursor missions (Weisbin et al., 2007). Though recent developments have delayed the target date for a manned return to the Moon, such missions are inevitable in the world's space exploration future. Much like a construction project on Earth, detailed maps will be required to conduct operations such as site selection, site preparation, and base construction. Similarly, precise localization will be necessary to facilitate load transport, scientific investigation, and in-situ resource utilization. The combination of these tasks will be essential to operations in the vicinity of a lunar base. In contrast to the exploratory traverses conducted by the Mars Exploration Rovers (MERs), these rovers will perform extended operations in local *worksite* environments, revisiting the same locations multiple times. As a result, accurate mapping and localization algorithms are key enabling technologies for autonomous worksite operations.

Though we use the lunar scenario for motivation, the design decisions made in this thesis apply to a variety of environments. If a base were to be established on the Moon, it is likely that it would be located in a crater near the Lunar South Pole (Dale, 2006). This crater could contain some permanently shadowed regions, which may hide frozen water (Weisbin et al., 2007). Preliminary results from the LCROSS mission have confirmed this hypothesis (Colaprete et al., 2010). Furthermore, investigation of the Lunar South Pole terrain could help validate some geological theories (Tera et al., 1974). The previously obtained samples from the equatorial region have been insufficient in answering these scientific queries, which prompts the need for polar exploration. In this case, the worksite might be defined by the boundaries of the crater.

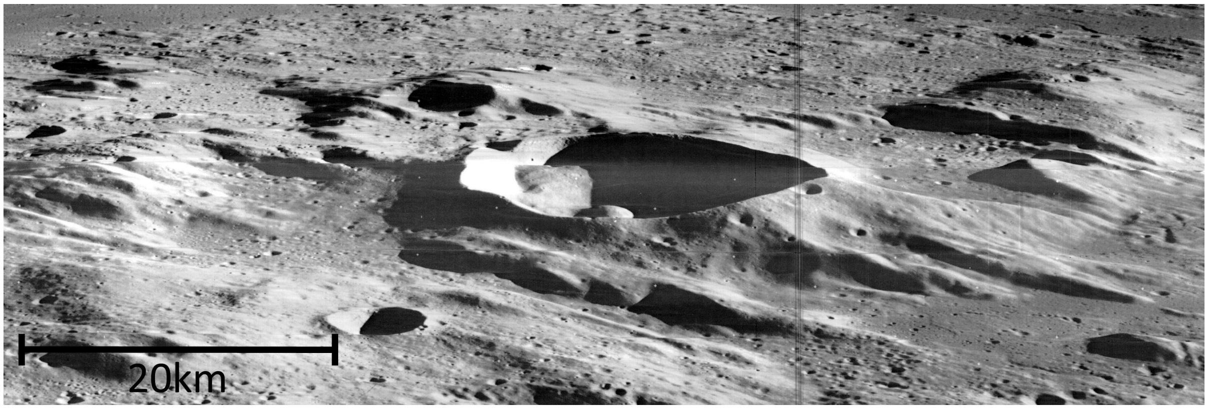


Figure 1.1: An image of Ukert crater on the lunar surface obtained during the Lunar Orbiter 3 mission. With poor lighting conditions and permanently shadowed regions, an actively illuminated sensor such as a laser rangefinder is appropriate for rover navigation. *Photo credit: NASA.*

However, the desire for continuous robotic operations in a planetary environment comes with a number of implications. Without an infrastructure-based positioning system such as GPS, onboard sensing must be used for navigation. Furthermore, operation in permanently shadowed regions restricts the choice of sensors. Finally, operations on planetary surfaces require systems that are able to handle the rugged, three-dimensional (3D) terrain, as depicted in Figure 1.1.

For 3D navigation in unstructured environments, many common implementations in the literature utilize passive cameras. In particular, impressive localization and mapping results have been demonstrated using a stereo camera, in both frame-to-frame Visual Odometry (VO) (Konolige et al., 2007), and the use of bundle adjustment (Konolige and Agrawal, 2008) for loop closure. Stereo camera VO has been used successfully on the MERs (Maimone et al., 2007) and the Mars Science Laboratory (MSL) (Johnson et al., 2008) for motion estimation, outperforming wheel odometry in areas with loose soil. However, future worksite mapping missions will be performed on a much larger scale, requiring continuous operations subject to the varying lighting conditions of a planetary surface. Unfortunately, the small field-of-view and reliance on external lighting make cameras unsuitable for this task. This assertion was confirmed by McManus et al. (2011), who demonstrated that the lighting change over the course of a day severely degrades the matching performance of visual feature detectors applied to camera imagery. While attaching a light source to a stereo camera rig should provide some lighting stability, Husmann and Pedersen (2008) found that image quality similar to daylight conditions were difficult to achieve. Furthermore, the power requirements for sufficient illumination suitable for dark-navigation were shown to be very restrictive (Pedersen et al., 2008). Alternatively, laser rangefinders are not subject to ambient lighting concerns, have greater range, and a larger field-of-view.

Motivated by these advantages and future applications of laser rangefinders, this thesis focuses on the state estimation algorithms necessary to enable autonomous mapping and navigation in planetary worksite environments. We utilize a rover-mounted panning laser rangefinder as our primary sensor, as depicted in Figure 1.2, and address the challenges that arise from this sensor modality. The mapping and navigation tasks are considered separately, since each task implies a different mode of operation.

The mapping task is approached as a surveying operation, where a *stop-scan-go* traverse is employed to obtain long-range, 360° scans at infrequent stops. For alignment, we take a sparse feature-based approach, where distinctive geometric peaks are identified from the scans, and utilized alongside odometry measurements in a 3D Simultaneous Localization and Mapping (SLAM) formulation to estimate both the rover poses and construct a dense point cloud map.

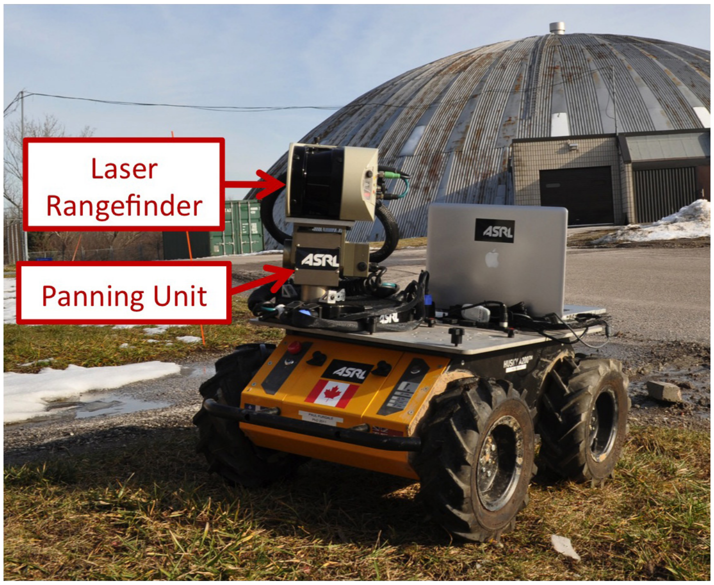


Figure 1.2: A Clearpath Husky A200 equipped with a laser rangefinder mounted on a panning unit. This is the primary sensor configuration used for mapping and navigation in this thesis.

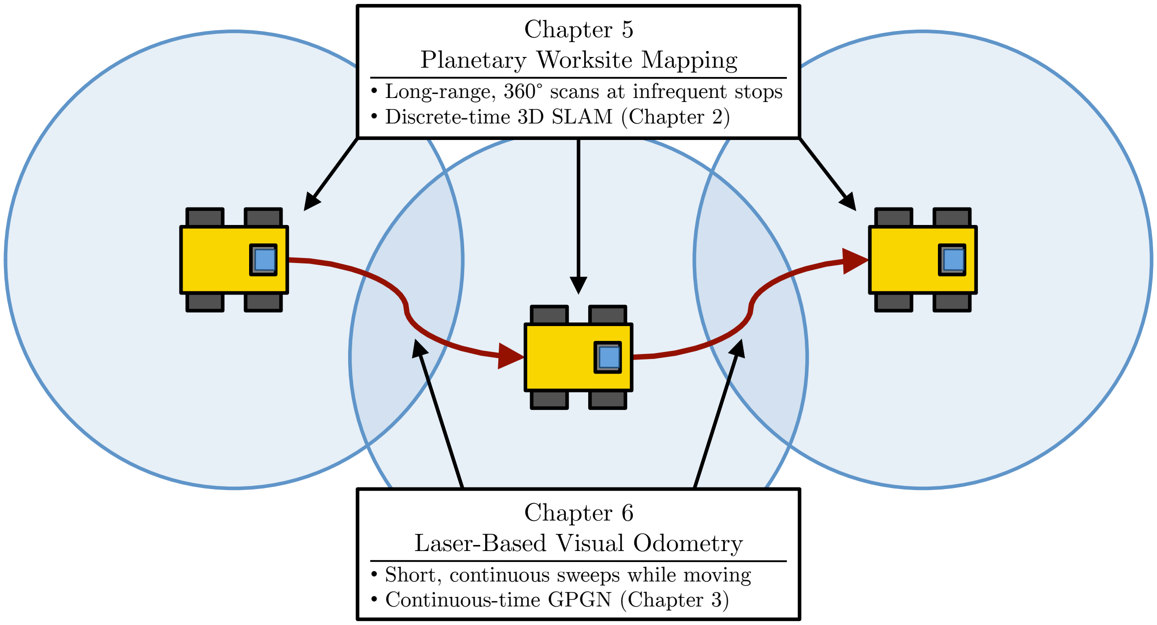

Chapter 5
Planetary Worksite Mapping
• Long-range, 360° scans at infrequent stops
• Discrete-time 3D SLAM (Chapter 2)
Chapter 6
Laser-Based Visual Odometry
• Short, continuous sweeps while moving
• Continuous-time GPGN (Chapter 3)

Figure 1.3: An illustration depicting the integration of the mapping and navigation components in a planetary worksite mapping scenario. The key features and relevant sections of each component are also indicated.

In contrast, navigation is addressed by configuring the same panning laser rangefinder to perform short, continuous sweeps while the rover is in motion. We employ a sparse appearance-based approach for motion estimation, where we construct images from laser intensity data, and compute VO estimates. Since the laser data is collected while the rover is moving, the intensity images are subject to motion distortion. These issues are overcome by considering the timestamps of the feature measurements, and by formulating the estimation problem in continuous time. This is facilitated by the introduction of Gaussian Process Gauss-Newton (GPGN) (Tong et al., 2012b, 2013b), a novel algorithm for nonparametric, continuous-time, nonlinear, batch state estimation. Experimental validation is provided for both mapping and navigation components using data gathered at multiple planetary analogue test sites.

The combination of these mapping and navigation frameworks provide the enabling technology for autonomous planetary worksite operations. Mapping aligns the long-range scans obtained at each stop, while navigation provides the necessary feedback to maneuver the rover between stops. This interplay between the algorithms and sensor configurations is illustrated in Figure 1.3.

This thesis contains a number of novel elements. The major highlights are the theoretical contribution in

1. The introduction of GPGN, an algorithm for nonparametric, continuous-time, nonlinear, batch state estimation.

and the algorithmic frameworks with unique features, which are summarized as follows:

2. A laser-based 3D SLAM framework that utilizes a stop-scan-go traverse in combination with hybrid data association, batch alignment, heterogeneous outlier rejection, and post-alignment verification to create robust, accurate worksite maps of natural, unstructured terrain.
3. The formulation of a laser-based VO algorithm for navigation that employs a continuously sweeping laser rangefinder, extracts sparse features from laser intensity images, and addresses state estimation using GPGN.

The structure of the thesis is depicted in Figure 1.4, where we have identified the dependencies between the chapters. We begin by reviewing the conventional method for discrete-time batch state estimation in Chapter 2. This is employed by our mapping framework, which addresses the SLAM problem by modelling the stop-scan-go traverse as discrete poses. We follow with the development of our novel nonparametric, continuous-time batch state estimation algorithm, GPGN, in Chapter 3. This formulation is necessary for navigation due to the motion distortion effects caused by collecting sensor data while the rover platform is in motion.

As the state estimation algorithms are derived using linear algebra, some complexities arise when attempting to apply the techniques to 3D state estimation. The primary challenge stems from the nature of rotations, which conflicts with the vector space assumptions that underlie the algorithmic derivations. These issues are addressed in Chapter 4, where we detail the necessary modifications employed in this thesis to extend the estimation algorithms into the 3D domain.

These theoretical contributions culminate in the mapping framework in Chapter 5, which describes the algorithmic integration necessary to create accurate maps of planetary worksite terrain autonomously. This is followed by our laser-based VO algorithm in Chapter 6, which makes use of intensity images, sparse appearance-based features, and GPGN to estimate the rover motion.

Finally, a summary of the contributions, discussion on future work, and concluding thoughts are provided in Chapter 7.

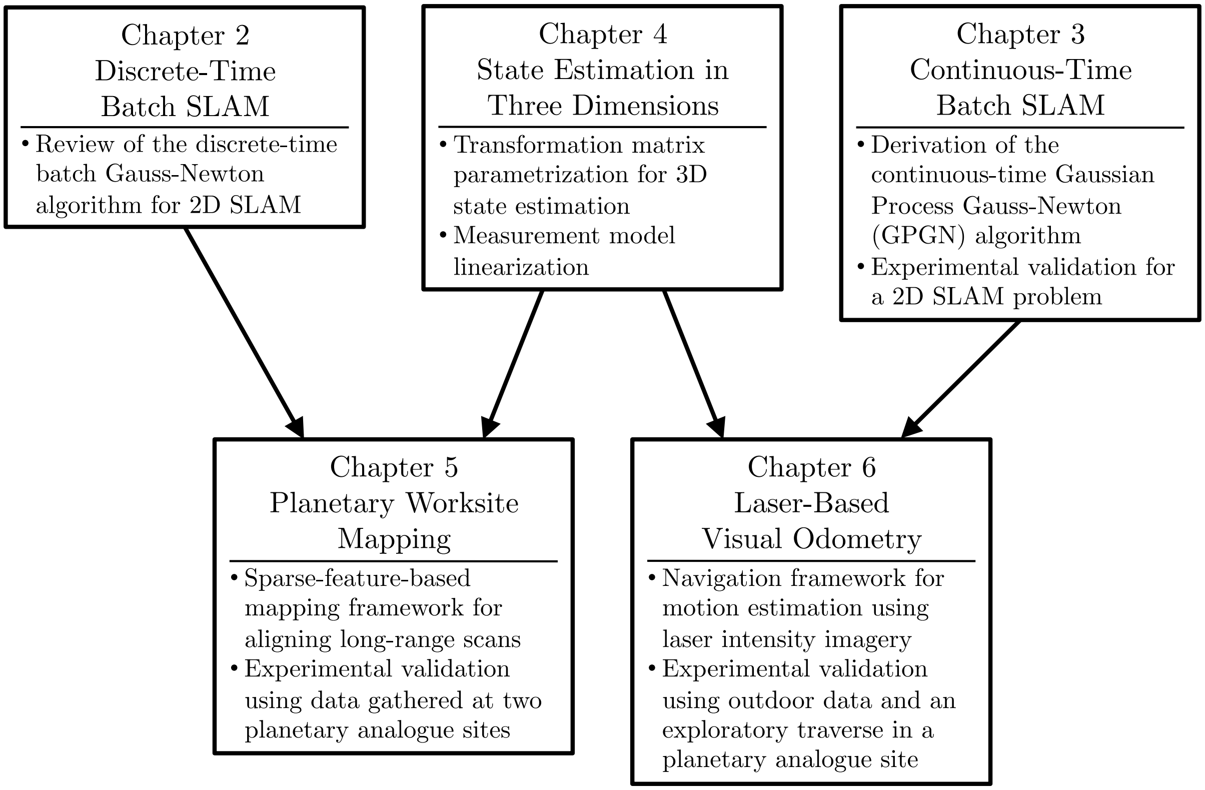

Chapter 2
Discrete-Time
Batch SLAM
• Review of the discrete-time batch Gauss-Newton algorithm for 2D SLAM
Chapter 4
State Estimation in
Three Dimensions
• Transformation matrix parametrization for 3D state estimation
• Measurement model linearization
Chapter 3
Continuous-Time
Batch SLAM
• Derivation of the continuous-time Gaussian Process Gauss-Newton (GPGN) algorithm
• Experimental validation for a 2D SLAM problem
Chapter 5
Planetary Worksite
Mapping
• Sparse-feature-based mapping framework for aligning long-range scans
• Experimental validation using data gathered at two planetary analogue sites
Chapter 6
Laser-Based
Visual Odometry
• Navigation framework for motion estimation using laser intensity imagery
• Experimental validation using outdoor data and an exploratory traverse in a planetary analogue site

Figure 1.4: A diagram depicting the highlights and dependencies between chapters in this thesis.

Chapter 2

Discrete-Time Batch SLAM

2.1 Introduction

In this chapter, we review the conventional method for discrete-time batch state estimation. This is the standard formulation for the majority of state estimation problems in the mobile robotics literature, and it is the core estimation method utilized in our mapping framework described in Chapter 5.

We consider the traditional vector space derivation, and discuss the modifications necessary for extension to 3D in Chapter 4. Though this approach generalizes to other state estimation problems, we focus on the SLAM scenario to reflect the intended mapping application. In this problem, a robot begins in unknown terrain, and travels to different locations while obtaining measurements from a nonlinear, noninvertible sensor model at discrete times. We take a sparse feature-based approach to this problem, where our map is composed of distinct features in the scene.

This common scenario is typically addressed by discretizing the robot trajectory, and computing a rover pose estimate at each measurement instant. As a result, we seek to determine values for the discrete robot poses and the landmark positions. This is illustrated in Figure 2.1. This is an appropriate model for our mapping scenario, since the robot stops at discrete locations to gather sensor data.

The conventional approach to solving the 2D discrete-time problem was established in the seminal paper by Smith et al. (1990), who performed probabilistic estimation using a combined state vector consisting of both the robot poses and landmark positions. This formulation of the SLAM problem as a nonlinear state estimation problem spurred the development of a wide range of algorithmic solutions.

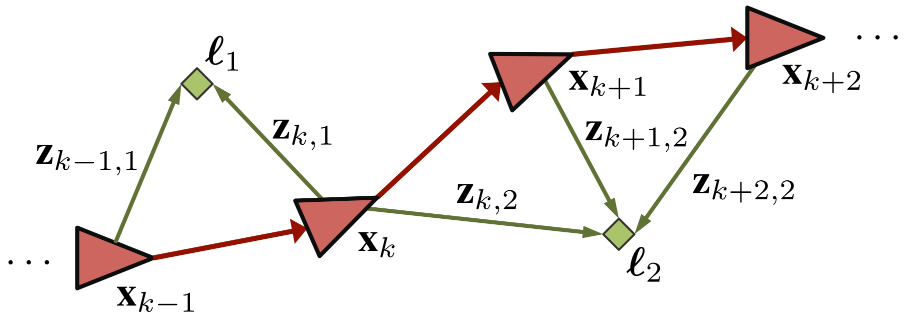


Figure 2.1: An illustration of the conventional discrete-time state estimation formulation, with robot poses depicted by red triangles, landmarks by green diamonds, pose ordering by red arrows, and measurements by green arrows. In this formulation, the robot poses are discretized and enumerated in sequence, and the measurement timestamps are discarded. This matches the stop-scan-go nature of the mapping scenario.

Initial solutions involved filters such as the Extended Kalman Filter (EKF) (Kalman, 1960), Sigma-Point Kalman Filter (Julier and Uhlmann, 1997), and the Particle Filter (Thrun et al., 2001). However, it was found that these algorithms can produce inconsistent estimates due to the Markov assumption (Julier and Uhlmann, 2001; Bailey et al., 2006).

In a parallel thread, Lu and Milios (1994) addressed the consistency issues by considering the measurements as a batch, and utilized optimization methods for state estimation. Through extensions of these concepts with algorithms such as GraphSLAM (Thrun and Montemerlo, 2006), 6DSLAM (Borrmann et al., 2008), and related developments in bundle adjustment (Brown, 1958) from the computer vision literature, it was found that the batch approach also provided greater accuracy due to iteratively refined linearization (Bell and Cathey, 1993). Furthermore, numerous researchers (Thrun and Montemerlo, 2006; Dellaert and Kaess, 2006; Kaess et al., 2008, 2012; Strasdat et al., 2012) have shown that the batch approach is computationally tractable given modern computing hardware, and should be utilized when possible.

2.2 Related Work

While there are many variants and substantial advances available in the literature concerning batch estimation, we focus on the basic formulation in this chapter. In particular, we note the fact that the Gauss-Newton nonlinear optimization algorithm (Gauss, 1855) sits at the core of the majority of batch state estimation algorithms. For example, it is used in bundle adjustment (Brown, 1958), GraphSLAM (Thrun and Montemerlo, 2006), 6DSLAM (Borrmann et al., 2008), and MOGA (Carle et al., 2010). Numerous software packages are also available that implement Gauss-Newton for pose graph relaxation and batch SLAM, such as g^{2}o (Kümmerle et al., 2011), and GTSAM[1].

All of these algorithms share the same underlying approach. The nonlinear estimation task is formulated as an optimization problem, where a Maximum Likelihood objective function is utilized, and an iterative approach is taken to find the optimal solution. These methods differ only by their state parametrization when conducting 3D estimation. This is discussed in further detail in Chapter 4.

2.3 Discrete-Time Batch Gauss-Newton

Our derivation for the discrete-time batch Gauss-Newton algorithm for state estimation begins with a brief mathematical review of the core probabilistic concepts, and the definition of our state estimation problem. This is followed by the optimization algorithm formulation that is utilized to determine the state estimate, and concluded with a summary of the steps of the algorithm.

2.3.1 Background

The most commonly used discrete-time probability distribution in mobile robotics is the Gaussian random variable. A Gaussian random variable, $\mathbf{x}$, is expressed as

$$\mathbf{x} \sim \mathcal{N}(\mathbf{m}, \mathbf{K}), \tag{2.1}$$

where $\mathbf{m}$ is the mean vector, and $\mathbf{K}$ is the covariance matrix. While there are some statistical theories that justify its use in modelling process and sensor noise such as the Central Limit Theorem, the key advantage in using this probability distribution is the fact that it is completely parametrized by two quantities. This provides a compact representation for the distribution, and allows for simple algebraic operations. As a result, the Gaussian assumption is employed throughout the state estimation literature.

[1] https://collab.cc.gatech.edu/borg/gtsam/

2.3.2 Problem Statement

In the SLAM scenario, we are interested in estimating both the robot poses, as well as the landmark positions. While functionally equivalent, we explicitly divide the state into two components to highlight the structure of the problem, as well as to provide a parallel to the continuous-time derivation presented in Chapter 3.

Our state, $\boldsymbol{\theta}$, is defined to be

$$\boldsymbol{\theta} := \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\ell} \end{bmatrix} \tag{2.2}$$

where $\mathbf{x}$ is a vector containing discrete robot poses, and $\boldsymbol{\ell}$ is a vector containing the landmark positions. We consider a state-space measurement model of the form

$$\mathbf{z}_i := \mathbf{h}_i(\boldsymbol{\theta}) + \mathbf{n}_i, \ \mathbf{n}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_i), \tag{2.3}$$

where the measurements, $\mathbf{z}_i$, are obtained through a nonlinear, noninvertible measurement model, $\mathbf{h}_i(\cdot)$, and for simplicity, we have modelled the measurement noise, $\mathbf{n}_i$, as additive, zero-mean, and Gaussian with covariance $\mathbf{R}_i$. In the SLAM scenario, this measurement model typically involves a single robot pose and a single landmark for feature measurements, or pairs of robot poses in the case of odometry.

The goal of the batch state estimation problem is to determine the probability distribution of the state given a collection of N measurements. Following the standard practice in mobile robotics, we maintain a Gaussian assumption and only estimate the first two moments of the probability distribution. To accomplish this, we cast this estimation task as an optimization problem. The standard approach utilizes a sum-of-squares objective function of the form $J := \frac{1}{2}\sum_{i=1}^{N} \mathbf{e}_i^T \mathbf{W}_i \mathbf{e}_i$, where $\mathbf{e}_i$ is an error term, and $\mathbf{W}_i$ is its associated weight. If the weight is chosen to be the inverse covariance matrix of $\mathbf{e}_i$, a Mahalanobis distance (Mahalanobis, 1936) is formed. That is, the objective function we seek to minimize is

$$J := \frac{1}{2} \sum_{i=1}^{N} \left(\mathbf{z}_i - \mathbf{h}_i(\boldsymbol{\theta})\right)^T \mathbf{R}_i^{-1} \left(\mathbf{z}_i - \mathbf{h}_i(\boldsymbol{\theta})\right). \tag{2.4}$$

Minimizing this objective function produces the Maximum Likelihood solution.

2.3.3 Optimization

To find the minimum of this objective function, we take the Gauss-Newton approach (Gauss, 1855) of linearizing the error terms, minimizing the resulting quadratic function, and iterating until convergence. We linearize the measurement model by making the assumption that the state is approximated by the value of the current estimate, $\overline{\boldsymbol{\theta}}$, and an additive perturbation, $\delta\boldsymbol{\theta}$. That is,

$$\boldsymbol{\theta} \approx \overline{\boldsymbol{\theta}} + \delta\boldsymbol{\theta}. \tag{2.5}$$

At each iteration, we seek the optimal value of the perturbation, $\delta\boldsymbol{\theta}^\star$, which we apply to bring our estimate progressively closer to the optimal value of the state. Under this assumption, our linearized measurement model is

$$\mathbf{z}_i \approx \mathbf{h}_i(\overline{\boldsymbol{\theta}}) + \mathbf{H}_i\,\delta\boldsymbol{\theta} + \mathbf{n}_i, \;\; \mathbf{H}_i := \left.\frac{\partial \mathbf{h}_i}{\partial \boldsymbol{\theta}}\right|_{\overline{\boldsymbol{\theta}}}. \tag{2.6}$$

Since the conventional discrete-time state estimation formulation discretizes the robot pose at each measurement time, we require these estimates to linearize the measurements. As a result, all state values are updated and stored at each iteration.

Next, we simplify the objective function by defining

$$\mathbf{z} := \begin{bmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_N \end{bmatrix}, \; \mathbf{h} := \begin{bmatrix} \mathbf{h}_1(\overline{\boldsymbol{\theta}}) \\ \vdots \\ \mathbf{h}_N(\overline{\boldsymbol{\theta}}) \end{bmatrix}, \; \mathbf{H} := \begin{bmatrix} \mathbf{H}_1 & & \\ & \ddots & \\ & & \mathbf{H}_N \end{bmatrix}, \; \mathbf{R} := \begin{bmatrix} \mathbf{R}_1 & & \\ & \ddots & \\ & & \mathbf{R}_N \end{bmatrix}, \tag{2.7}$$

which results in

$$J \approx \frac{1}{2}\left(\mathbf{z} - \mathbf{h} - \mathbf{H}\delta\boldsymbol{\theta}\right)^T \mathbf{R}^{-1}\left(\mathbf{z} - \mathbf{h} - \mathbf{H}\delta\boldsymbol{\theta}\right). \tag{2.8}$$

Taking the derivative with respect to $\delta\boldsymbol{\theta}$ gives us

$$\frac{\partial J}{\partial \delta\boldsymbol{\theta}}^T = -\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{z} - \mathbf{h} - \mathbf{H}\delta\boldsymbol{\theta}), \tag{2.9}$$

and setting the value to zero provides us with $\delta\boldsymbol{\theta}^\star$, the optimal value of the perturbation that minimizes the quadratic expression at this iteration. This can be rearranged to express the mean and the covariance of $\delta\boldsymbol{\theta}^\star$ in information form as

$$\underbrace{\left(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\right)}_{=:\mathbf{A}}\delta\boldsymbol{\theta}^\star = \underbrace{\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{z} - \mathbf{h})}_{=:\mathbf{b}}, \tag{2.10}$$

$$\operatorname{cov}(\delta\boldsymbol{\theta}^\star, \delta\boldsymbol{\theta}^\star)^{-1} = \mathbf{A}. \tag{2.11}$$

Although we can solve the system of equations in Equation (2.10) directly through matrix inversion, the structure of the SLAM problem can be exploited to produce a more efficient algorithm. This structure, depicted in Figure 2.2, can be seen if we partition the linear system by the robot pose and landmark variables as

$$\underbrace{\left[\begin{array}{c|c}\mathbf{A}_{xx} & \mathbf{A}_{x\ell} \\ \hline \mathbf{A}_{x\ell}^T & \mathbf{A}_{\ell\ell}\end{array}\right]}_{\mathbf{A}}\underbrace{\left[\begin{array}{c}\delta\mathbf{x}^\star \\ \hline \delta\boldsymbol{\ell}^\star\end{array}\right]}_{\delta\boldsymbol{\theta}^\star} = \underbrace{\left[\begin{array}{c}\mathbf{b}_x \\ \hline \mathbf{b}_\ell\end{array}\right]}_{\mathbf{b}}, \tag{2.12}$$

which produces an arrowhead matrix (Brown, 1958). This type of system can be efficiently solved using the Schur complement. Premultiplying both sides of (2.12) by

$$\left[\begin{array}{c|c}\mathbf{1} & -\mathbf{A}_{x\ell}\mathbf{A}_{\ell\ell}^{-1} \\ \hline \mathbf{0} & \mathbf{1}\end{array}\right] \tag{2.13}$$

results in

$$\left[\begin{array}{c|c}\mathbf{A}_{xx} - \mathbf{A}_{x\ell}\mathbf{A}_{\ell\ell}^{-1}\mathbf{A}_{x\ell}^T & \mathbf{0} \\ \hline \mathbf{A}_{x\ell}^T & \mathbf{A}_{\ell\ell}\end{array}\right]\left[\begin{array}{c}\delta\mathbf{x}^\star \\ \hline \delta\boldsymbol{\ell}^\star\end{array}\right] = \left[\begin{array}{c}\mathbf{b}_x - \mathbf{A}_{x\ell}\mathbf{A}_{\ell\ell}^{-1}\mathbf{b}_\ell \\ \hline \mathbf{b}_\ell\end{array}\right]. \tag{2.14}$$

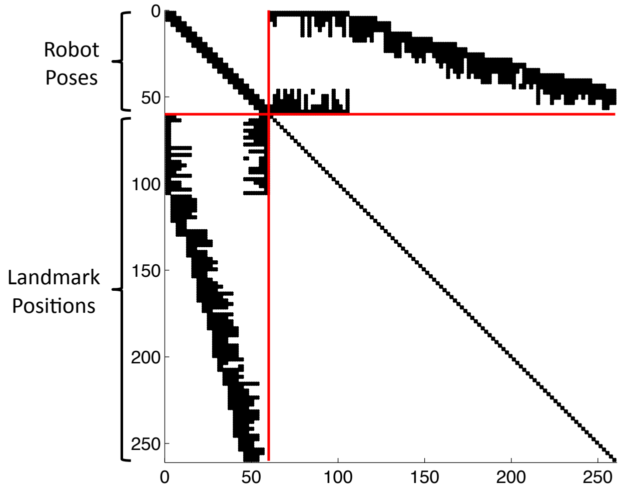


Figure 2.2: Plot depicting the sparsity pattern of the inverse covariance matrix, $\mathbf{A}$ (2.12), for the discrete-time batch Gauss-Newton formulation of a sample 2D SLAM problem involving 20 robot poses and 100 landmarks. The red lines indicate the partitioning between the robot poses and landmark positions, and the landmarks are sorted in order of first observation. Linear systems of this form can be solved efficiently using the Schur complement.

This system of equations has the same solution as (2.12), but we can exploit the fact that $\mathbf{A}_{\ell\ell}$ is easily inverted due to its block-diagonal form. The robot pose variables, $\delta\mathbf{x}^{\star}$, can be determined directly through (2.14), and the remaining landmark position variables, $\delta\boldsymbol{\ell}^{\star}$, can be found inexpensively through back-substitution. This method is known as sparse bundle adjustment (SBA) in the computer vision literature (Brown, 1958; Hartley and Zisserman, 2000), and can be interpreted as conditioning the robot poses on the landmark positions (Thrun and Montemerlo, 2006).

If we let K be the number of poses and L be the number of landmarks, the naive matrix inversion method has a computational complexity of $O((K+L)^3)$. Application of the Schur complement to condition the robot poses on the landmark positions reduces the complexity to $O(K^3 + K^2L)$.

After solving for $\delta\boldsymbol{\theta}^{\star}$, it is applied as an update to the current state estimate by $\overline{\boldsymbol{\theta}} \leftarrow \overline{\boldsymbol{\theta}} + \delta\boldsymbol{\theta}^{\star}$. The system is then relinearized using the improved state estimate, and the process repeats until convergence. In practice, augmentations may be needed to improve the robustness and convergence properties, such as line searches, M-estimation (Zhang, 1997), and Levenberg-Marquardt regularization (Levenberg, 1944; Marquardt, 1963).

2.3.4 Summary

In summary, discrete-time batch state estimation using the Gauss-Newton algorithm is performed as follows:

1. Initialize the algorithm with an initial guess for the state, $\overline{\boldsymbol{\theta}}$.
2. Linearize and construct the (2.7) matrices.
3. Solve for the optimal perturbations, $\delta\boldsymbol{\theta}^{\star}$, using (2.10).
4. Apply the additive update to improve the state estimate, $\overline{\boldsymbol{\theta}} \leftarrow \overline{\boldsymbol{\theta}} + \delta\boldsymbol{\theta}^{\star}$.
5. Repeat steps 2-4 until convergence.
6. After convergence, compute the covariance of the state estimate using (2.11).

Chapter 3

Continuous-Time Batch SLAM

3.1 Introduction

In this chapter, we present an alternative formulation that considers the batch state estimation problem in continuous time. While the discrete-time approach has served the robotics community well in the past, this continuous-time formulation enables novel applications such as our laser-based VO framework presented in Chapter 6.

Once again, we present a vector space derivation in this chapter, and discuss the modifications necessary for 3D implementation in Chapter 4. We also utilize the SLAM problem as our illustrative example, because this problem is well-understood, and provides opportunities to draw parallels to the existing discrete-time techniques. We motivate this novel formulation by focusing on the fact that in contrast to the stop-scan-go mode of operation employed for mapping, the robot is travelling continuously during navigation.

As stated in the previous chapter, the typical formulation employed in mobile robotics discretizes the trajectory and discards the timestamps. This formulation is depicted in Figure 2.1. While the discrete-time approach appropriately models sensors where multiple measurements are captured at discrete instants of time, it does not reflect the true physical reality of the underlying system. In reality, the robot motion is continuous, with characteristics defined by the physical and temporal constraints of the mobile platform. Therefore, we propose to model the robot trajectory in continuous time, as depicted in Figure 3.1.

In addition to modelling the continuous trajectory improperly, the discrete-time approach of computing an estimate at every measurement instant can be restrictive. This is because a noninvertible measurement model imposes the requirement of multiple sensor

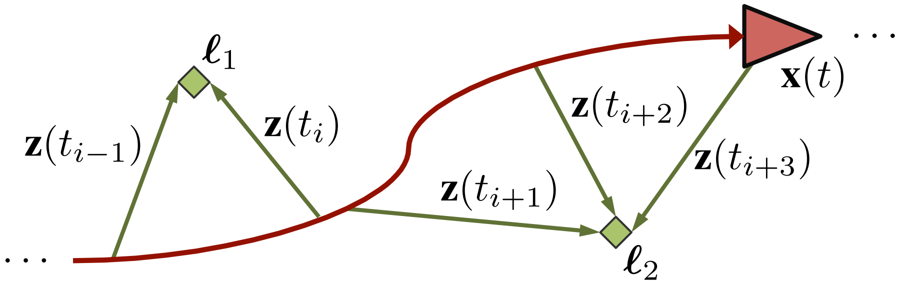


Figure 3.1: An illustration of the proposed continuous-time state estimation formulation, with the robot trajectory depicted by a smooth red line and triangle, landmarks by green diamonds, and measurements by green arrows. In this formulation, the robot pose is modelled as a smooth function of time, and the discrete measurements are stored along with their timestamps. Furthermore, it can be seen that asynchronous measurements can be handled in the same manner.

readings to produce an estimate for every robot pose. This requirement is problematic for high-rate sensors such as inertial measurement units, rolling shutter cameras, or scanning laser systems, which tend to produce measurements at unique and asynchronous times. The problem is intensified if we desire additional state estimates at instants between measurement times. For example, the dense data from a scanning laser rangefinder could be used for accurate terrain modelling, if we were able to produce a pose estimate for every point. These issues of measurement scarcity are addressed by continuous-time state estimation, because the underlying continuous-time process model provides sufficient information to produce an estimate at any instant of time.

In the past literature, the traditional discrete-time Extended Kalman Filter (EKF) (Smith et al., 1990) has also been formulated in continuous time (Kalman and Bucy, 1961; Jazwinski, 1970). Unfortunately, these filtering methods require numerical integration for practical implementation, and can produce jagged estimates due to the Markov assumption. Similar implementation issues arise for forward/backward smoothing methods (Simon, 2006). Once again, we would like to take advantage of modern computing hardware by taking a batch approach.

To develop a new method of continuous-time batch state estimation in a framework amenable to robotics, we look to current work from the machine learning community addressing the *regression problem*. The regression problem considers the case where sample outputs are obtained from an unknown continuous function, and the goal is to predict the value of the function for a new input. Additional complexity arises when the sample outputs have corruptive noise, which requires probabilistic approaches. The state estimation problem has a similar form, where we start with noisy measurements and seek to estimate states at given instants of time. However, instead of making further measurement predictions, we seek to determine the latent states that generated the measurements. In the SLAM example, the latent states are the robot poses and landmark locations. This task is complicated by the presence of nonlinear, noninvertible sensor models.

Solutions to the regression problem can be divided into two categories: parametric and nonparametric. In parametric regression, the unknown function is assumed to have a particular form, which is defined by a set of parameters. For example, the unknown function can be modelled as a polynomial, or the weighted sum of a set of known basis functions. With this assumed model, the regression problem reduces to finding the optimal parameter values that produce the best matches to the sample outputs. These optimized parameters can then be used for predicting new outputs. This is an effective method, but its accuracy is highly dependent on model selection. For parametric regression, the computational complexity is related to the number of chosen parameters.

The discrete-time batch state estimation formulation shares some similarities with the parametric regression approach, since the robot state is parametrized by discrete poses at instants of time (Figure 2.1). Determination of the optimal parameter values addresses the state estimation problem. This approach can be extended to the continuous-time domain by utilizing a continuous state representation, such as piecewise splines (Bibby and Reid, 2010; Furgale et al., 2012). However, the performance still suffers from the same issues of modelling accuracy.

In this chapter, we present an alternative approach based on nonparametric regression. Rather than assuming a parametric form, nonparametric techniques allow the unknown function to implicitly lie in a set of functions, which offers significantly more representational ability. However, instead of producing a model from the data, the implicit models are resolved by using the data itself to produce predictions. As a result, the computational complexity scales with the number of measurements.

Gaussian Process (GP) regression is a nonparametric method where the underlying function is modelled by a mean and covariance function (Rasmussen and Williams, 2006). This probabilistic representation accounts for uncertainty in the observations, and natively suppresses model complexity to avoid overfitting. As a result, it is a suitable candidate for adaptation to the continuous-time batch state estimation problem. Unfortunately, this is not a straightforward task, because we are interested in estimating the latent states, and the measurement models are nonlinear and noninvertible. We accomplish our task by modelling the robot state directly as a GP with the input of time, and utilize the Gauss-Newton algorithm (Gauss, 1855) to recover the latent states from the measurements.

In summary, we introduce the *Gaussian Process Gauss-Newton* (GPGN) algorithm in this chapter. GPGN is a nonparametric, continuous-time, nonlinear, batch state estimation algorithm, which utilizes concepts from GP regression applied to the robotics domain. This is a novel application of GPs, which is significantly different from the previous uses of GPs in robotics. In particular, we present an algorithm that can be applied to conventional state estimation scenarios. Parallels to the discrete-time approach are drawn where possible, and the advantages are highlighted through experimental illustration. GPGN should be considered as an alternative to the parametric approach of Furgale et al. (2012) for solving the continuous-time batch state estimation problem.

The contributions of this chapter have appeared in a number of publications. GPGN was first introduced as a state estimation algorithm in Tong et al. (2012b), where the weight-space derivation was utilized to illustrate the parallels to the parametric approach. This was followed by Tong et al. (2013b), which extended GPGN to the SLAM scenario, and presented the simplified function-space approach for completeness.

The remainder of this chapter is organized as follows. We begin with a discussion of related applications of GPs to robotic state estimation in Section 3.2. This is followed by the derivation of GPGN in Section 3.3, and experimental validation in Section 3.4. Finally, concluding remarks are provided in Section 3.5.

3.2 Related Work

The application of GPs to the field of robotics is not a new concept. GPs have been applied successfully to learn measurement models for complex systems such as laser rangefinders (Plagemann et al., 2007), map gas distributions (Stachniss et al., 2009), and perform large-scale terrain modelling (Vasudevan et al., 2009). In fact, GP terrain modelling can be traced back to the field of geostatistics, where it was termed *kriging* (Matheron, 1963). These applications utilized GPs in their original formulation, since GPs were employed for modelling sensor outputs. Recent work by Guizilini and Ramos (2012) has adapted this formulation to the state estimation problem by using GPs to learn the mapping from optical flow vectors to metric visual odometry estimates.

GP measurement models have been utilized for state estimation via particle filters (Ferris et al., 2006), and Bayesian filtering with GP-BayesFilters (GPBFs) (Ko and Fox, 2009). The use of GP process and measurement models were also recently generalized into a discrete-time recursive filtering and smoothing framework by Deisenroth et al. (2012). Our work differs greatly from these approaches. Rather than applying GPs to first model the sensors and then utilize them for discrete-time filtering, we use GPs to perform the state estimation itself. That is, we do not model the sensors with GPs; we model the state itself with a GP, and utilize a batch estimation framework.

The concept of GP Latent Variable Models (GPLVMs) (Lawrence, 2003) has also been applied to the robotics domain. GPLVMs were originally developed as a method for dimensionality reduction, which mapped high-dimensional GP measurements to a lower-dimensional GP latent state. This concept was employed for Wifi-based Simultaneous Localization and Mapping (WifiSLAM) (Ferris et al., 2007) and human motion tracking (Wang et al., 2008). In addition, GPLVMs were used to train GPBFs without ground truth measurements (Ko and Fox, 2011). However, GPLVMs require a number of assumptions for successful performance in the robotics domain. Since GPLVMs were originally conceived as a dimensionality reduction technique, the assumption that similar sensor measurements are obtained from similar locations is required. This assumption does not hold when utilizing noninvertible measurement models. Furthermore, the lower-dimensional representation produced does not guarantee any relevance to the scenario at hand. To address this issue, additional weak labels are required to guide the process to produce physically meaningful results for the latent states, such as robot poses.

Though we also model the hidden state as a GP, in our approach, we retain the discrete-time sensor models. By restricting the modification to the estimator's internal representation of the state, we are able to address conventional state estimation scenarios, and established methods such as observability analysis still apply. Furthermore, the latent states represented by the GP retain their physical relevance.

Other related works in the machine learning literature include derivative GP observations (Solak et al., 2003), warped GPs (Snelson et al., 2004), and linear operator measurements (Särkkä, 2011), which share some similarities to how we utilize indirect observations of the state. A similar problem was considered by Archambeau et al. (2007), where the evolution of a Stochastic Differential Equation (SDE) was approximated with a GP based on discrete observations. However, the presented solution was restricted to simplified GPs that were parametrized by time-invariant coefficients. The inverse approach was also taken by Särkkä and Hartikainen (2012), which addressed the GP regression problem by converting the GP prior into an SDE, and performed Kalman filtering and smoothing for an online learning process. Our work differs from these approaches because we address the batch state estimation problem, utilize nonlinear measurement models, and apply the conventional Gauss-Newton optimization method (Gauss, 1855). In addition, we present our approach in a format more familiar to the robotics community.

3.3 Gaussian Process Gauss-Newton

In this section, we provide the derivations for GPGN. For generality, we consider a state composed of both a time-varying quantity, $\mathbf{x}(t)$, and a time-invariant component, $\boldsymbol{\ell}$. In the SLAM scenario, the time-varying quantity is the robot pose, and the time-invariant quantity is the landmark locations. Our task is to determine the values of the state given a set of N measurements obtained over a period of time.

We begin by providing a brief mathematical overview of GPs, and follow with the definition of the state estimation problem. Two derivations are provided in this section, reflecting both the weight-space and function-space approaches from the GP regression literature (Rasmussen and Williams, 2006). The weight-space formulation provides clear parallels to the discrete-time approach, but the function-space formulation is considerably more concise. In addition to the core derivations, an efficient expression for predicting the value of the time-varying state at additional times of interest is also developed, which leads to an approximation method that significantly reduces the computational requirements. The section is then concluded with a summary of the GPGN algorithm.

3.3.1 Background

In GPGN, Gaussian random variables are utilized alongside GPs. A GP can be considered as a generalization of a Gaussian random variable to the continuous-time domain (Rasmussen and Williams, 2006). Instead of a mean vector and a covariance matrix, a GP, $\mathbf{x}(t)$, is described by a mean *function*, $\boldsymbol{\mu}(t)$, and a covariance *function*, $\boldsymbol{\mathcal{K}}(t, t')$:

$$\mathbf{x}(t) \sim \mathcal{GP}(\boldsymbol{\mu}(t), \boldsymbol{\mathcal{K}}(t, t')). \tag{3.1}$$

These expressions are functions of time, and the covariance function involves two time variables to account for cross-temporal relations. Evaluating these expressions at discrete instants of time results in jointly Gaussian random variables.

3.3.2 Problem Statement

The state-space models for the underlying system are modelled as

$$\mathbf{x}(t) \sim \mathcal{GP}(\boldsymbol{\mu}(t), \boldsymbol{\mathcal{K}}(t,t')), \tag{3.2}$$

$$\boldsymbol{\ell} \sim \mathcal{N}(\mathbf{d}, \mathbf{L}) \tag{3.3}$$

$$\mathbf{z}_i = \mathbf{h}_i(\mathbf{x}(t_i), \boldsymbol{\ell}) + \mathbf{n}_i, \quad \mathbf{n}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_i), \tag{3.4}$$

where $\mathbf{x}(t)$ is a GP with mean and covariance functions $\boldsymbol{\mu}(t)$ and $\boldsymbol{\mathcal{K}}(t,t')$, respectively, $\boldsymbol{\ell}$ is a time-invariant discrete Gaussian random variable with prior mean of $\mathbf{d}$ and covariance $\mathbf{L}$, and the measurements, $\mathbf{z}_i$, are obtained through a conventional nonlinear, noninvertible measurement model, $\mathbf{h}_i(\cdot)$, at N discrete times, t_i. For simplicity, we have modelled the measurement noise, $\mathbf{n}_i$, as additive, zero-mean, and Gaussian with covariance $\mathbf{R}_i$.

In contrast to the initial step for GP regression where the additional parameters specifying the GP properties (known as *hyperparameters*) are first trained using the data, it should be noted that we do not address the specification of hyperparameters in our derivation of GPGN. Rather, we assume that they have already been obtained through some prior system identification process. In practice, these hyperparameters may be related to the physical properties of the underlying system, if an appropriate GP function is chosen. This assumption of prior characterization is typical for state estimation algorithms. For example, this assumption is applied for the measurement noise covariance, $\mathbf{R}_i$.

To simplify the expressions in the following sections, we begin by combining the two state components by defining

$$\boldsymbol{\theta}(t) := \begin{bmatrix} \mathbf{x}(t) \\ \boldsymbol{\ell} \end{bmatrix}, \quad \boldsymbol{\eta}(t) := \begin{bmatrix} \boldsymbol{\mu}(t) \\ \mathbf{d} \end{bmatrix}, \quad \boldsymbol{\mathcal{P}}(t,t') := \begin{bmatrix} \boldsymbol{\mathcal{K}}(t,t') & \mathbf{0} \\ \mathbf{0} & \mathbf{L} \end{bmatrix}, \tag{3.5}$$

which results in the simplified system equations:

$$\boldsymbol{\theta}(t) \sim \mathcal{GP}(\boldsymbol{\eta}(t), \boldsymbol{\mathcal{P}}(t,t')), \tag{3.6}$$

$$\mathbf{z}_i = \mathbf{h}_i(\boldsymbol{\theta}(t_i)) + \mathbf{n}_i. \tag{3.7}$$

It should be noted that we maintain the conventional assumption of independence between the robot poses and landmark position priors in the definition of $\boldsymbol{\mathcal{P}}(t,t')$. Correlations between the pose and landmark estimates arise when the measurements are incorporated. Using these models, we wish to determine the probability distribution of the state given a collection of N measurements, where we assume that the posterior state is modelled by a Gaussian process.

3.3.3 Weight-Space Derivation

The weight-space derivation begins with a parametric basis function representation for the state. This approach is used as an intermediate step, which allows us to illustrate the parallels to the conventional discrete-time approach by providing a familiar Gaussian random variable formulation. However, we do not utilize any basis functions in implementation, as algebraic manipulations are then conducted to produce a nonparametric form, resulting in the GPGN algorithm.

3.3.3.1 Basis Function Representation

Just like the discrete-time approach, we also utilize an objective function in our derivation. As an intermediate step, we begin by adopting a basis function representation for the state

$$\boldsymbol{\theta}(t) := \mathbf{\Psi}(t)\,\mathbf{b}, \tag{3.8}$$

where $\mathbf{\Psi}(t)$ are the basis functions and $\mathbf{b}$ are the coefficients defined by

$$\mathbf{\Psi}(t) := \begin{bmatrix} \mathbf{\Phi}(t) & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}, \quad \mathbf{b} := \begin{bmatrix} \mathbf{c} \\ \boldsymbol{\ell} \end{bmatrix}. \tag{3.9}$$

In this representation, the time-varying component of the state is replaced by $\mathbf{\Phi}(t)$, a stack of M known temporal basis functions,

$$\mathbf{\Phi}(t) := \begin{bmatrix} \boldsymbol{\phi}_1(t) & \dots & \boldsymbol{\phi}_M(t) \end{bmatrix}, \tag{3.10}$$

and $\mathbf{c}$, a column of coefficients. Though this representation would convert this problem into a parametric estimation problem (Furgale et al., 2012), we assume that the number of basis functions, M, is very large, or even infinite (Neal, 1994). This provides substantial representational power, but with the restriction that we cannot evaluate or store either $\mathbf{\Phi}(t)$ or $\mathbf{c}$. The nonparametric form that we will obtain in the following section avoids these issues.

With this representation, the measurement model becomes

$$\mathbf{z}_i = \mathbf{h}_i(\mathbf{\Psi}(t_i)\,\mathbf{b}) + \mathbf{n}_i, \tag{3.11}$$

and similarly, the mean and covariance functions of $\boldsymbol{\theta}(t)$ become

$$\boldsymbol{\eta}(t) =: \mathbf{\Psi}(t)\,\mathbf{u}, \tag{3.12}$$

$$\boldsymbol{\mathcal{P}}(t,t') =: \mathbf{\Psi}(t)\,\mathbf{P}\,\mathbf{\Psi}(t')^T, \tag{3.13}$$

where $\mathbf{u}$ and $\mathbf{P}$ are defined by

$$\mathbf{b} \sim \mathcal{N}(\mathbf{u}, \mathbf{P}). \tag{3.14}$$

For clarity, we keep the time-varying and time-invariant components separate by defining

$$\mathbf{u} =: \begin{bmatrix} \mathbf{m} \\ \mathbf{d} \end{bmatrix}, \quad \mathbf{P} =: \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & \mathbf{L} \end{bmatrix}. \tag{3.15}$$

These substitutions return us to the familiar Gaussian random variable domain, where (3.13) was obtained by invoking the 'kernel trick' from the machine learning literature (Rasmussen and Williams, 2006) in an inverse manner. This relation will be utilized later in Section 3.3.3.2 to resolve the issue of computational intractability due to the large number of basis functions.

We can then form a Maximum a Posteriori objective function:

$$J := \frac{1}{2} \sum_{i=1}^{N} \left(\mathbf{z}_i - \mathbf{h}_i(\mathbf{\Psi}(t_i)\,\mathbf{b})\right)^T \mathbf{R}_i^{-1} \left(\mathbf{z}_i - \mathbf{h}_i(\mathbf{\Psi}(t_i)\,\mathbf{b})\right) + \frac{1}{2} \left(\mathbf{b} - \mathbf{u}\right)^T \mathbf{P}^{-1} \left(\mathbf{b} - \mathbf{u}\right). \tag{3.16}$$

We continue by following the discrete-time Gauss-Newton optimization approach (Gauss, 1855) of linearizing the error terms, minimizing the resulting quadratic function, and iterating until convergence. Once again, we make the assumption that the state is approximated by the value of the current estimate, $\overline{\boldsymbol{\theta}}(t)$, and an additive perturbation, $\delta\boldsymbol{\theta}(t)$. Applying the basis function representation (3.8), we get

$$\begin{aligned} \boldsymbol{\theta}(t) &\approx \overline{\boldsymbol{\theta}}(t) + \delta\boldsymbol{\theta}(t) \\ &= \mathbf{\Psi}(t)\,\overline{\mathbf{b}} + \mathbf{\Psi}(t)\,\delta\mathbf{b}, \end{aligned} \tag{3.17}$$

where $\overline{\mathbf{b}}$ is our current estimate for the coefficients, and $\delta\mathbf{b}$ is the perturbation. At each iteration, we seek the optimal value of the perturbation, $\delta\mathbf{b}^\star$, which we apply to bring our estimate progressively closer to the optimal value of the state. Under this assumption, our linearized system models are

$$\overline{\mathbf{b}} + \delta\mathbf{b} \sim \mathcal{N}\left(\mathbf{u}, \mathbf{P}\right), \tag{3.18}$$

$$\mathbf{z}_i \approx \mathbf{h}_i(\overline{\boldsymbol{\theta}}(t_i)) + \mathbf{H}_i \mathbf{\Psi}(t_i)\,\delta\mathbf{b} + \mathbf{n}_i, \quad \mathbf{H}_i := \left.\frac{\partial \mathbf{h}_i}{\partial \boldsymbol{\theta}}\right|_{\overline{\boldsymbol{\theta}}(t_i)}, \tag{3.19}$$

and once again, we note that we require a value for the state at each measurement time to perform linearization. Since each measurement may come at a unique time, updating and storing these values can be prohibitive if there are a large number of measurements. This concern motivates the approximation method developed later in Section 3.3.6.

Finally, to simplify the objective function, we define

$$\mathbf{z} := \begin{bmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_N \end{bmatrix}, \qquad \mathbf{h} := \begin{bmatrix} \mathbf{h}_1(\overline{\boldsymbol{\theta}}(t_1)) \\ \vdots \\ \mathbf{h}_N(\overline{\boldsymbol{\theta}}(t_N)) \end{bmatrix}, \qquad \mathbf{H} := \begin{bmatrix} \mathbf{H}_1 & & \\ & \ddots & \\ & & \mathbf{H}_N \end{bmatrix},$$
$$\boldsymbol{\Psi} := \begin{bmatrix} \boldsymbol{\Phi} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}, \qquad \boldsymbol{\Phi} := \begin{bmatrix} \boldsymbol{\Phi}(t_1) \\ \vdots \\ \boldsymbol{\Phi}(t_N) \end{bmatrix}, \qquad \mathbf{R} := \begin{bmatrix} \mathbf{R}_1 & & \\ & \ddots & \\ & & \mathbf{R}_N \end{bmatrix}, \tag{3.20}$$

which results in

$$J = \frac{1}{2} \left(\mathbf{z} - \mathbf{h} - \mathbf{H}\,\boldsymbol{\Psi}\,\delta\mathbf{b}\right)^T \mathbf{R}^{-1} \left(\mathbf{z} - \mathbf{h} - \mathbf{H}\,\boldsymbol{\Psi}\,\delta\mathbf{b}\right) + \frac{1}{2} \left(\overline{\mathbf{b}} + \delta\mathbf{b} - \mathbf{u}\right)^T \mathbf{P}^{-1} \left(\overline{\mathbf{b}} + \delta\mathbf{b} - \mathbf{u}\right). \tag{3.21}$$

This expression is in the same form as the conventional discrete-time Gauss-Newton objective function. As a result, we proceed in the usual manner. Taking the derivative with respect to $\delta\mathbf{b}$ gives us

$$\frac{\partial J}{\partial\,\delta\mathbf{b}}^T = \left(-\mathbf{H}\boldsymbol{\Psi}\right)^T \mathbf{R}^{-1} \left(\mathbf{z} - \mathbf{h} - \mathbf{H}\,\boldsymbol{\Psi}\,\delta\mathbf{b}\right) + \mathbf{P}^{-1} \left(\overline{\mathbf{b}} + \delta\mathbf{b} - \mathbf{u}\right), \tag{3.22}$$

and setting the value to zero provides us with $\delta\mathbf{b}^\star$, the optimal value of the perturbation that minimizes the quadratic expression (at this iteration):

$$\underbrace{\left(\boldsymbol{\Psi}^T\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\boldsymbol{\Psi} + \mathbf{P}^{-1}\right)}_{=:\mathbf{A}} \delta\mathbf{b}^\star = \boldsymbol{\Psi}^T\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{z} - \mathbf{h}) - \mathbf{P}^{-1}\left(\overline{\mathbf{b}} - \mathbf{u}\right). \tag{3.23}$$

As can be seen, we now have correlations between the poses and the landmarks due to the landmark measurements. Since this derivation follows the Gauss-Newton approach, we can express the mean, $\delta\mathbf{b}^\star$, as

$$\delta\mathbf{b}^\star = \mathbf{A}^{-1}\left(\boldsymbol{\Psi}^T\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{z} - \mathbf{h}) - \mathbf{P}^{-1}(\overline{\mathbf{b}} - \mathbf{u})\right), \tag{3.24}$$

and the covariance as

$$\operatorname{cov}(\delta\mathbf{b}^\star, \delta\mathbf{b}^\star) = \mathbf{A}^{-1}. \tag{3.25}$$

These expressions provide the iterative solution to the parametric estimation problem.

3.3.3.2 Returning to a Nonparametric Form

We return to a nonparametric form by first applying the basis function relation $\delta\boldsymbol{\theta}^\star(t) = \boldsymbol{\Psi}(t)\,\delta\mathbf{b}^\star$, which produces the mean function

$$\delta\boldsymbol{\theta}^\star(t) = \boldsymbol{\Psi}(t)\,\mathbf{A}^{-1}\boldsymbol{\Psi}^T\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{z} - \mathbf{h}) - \boldsymbol{\Psi}(t)\,\mathbf{A}^{-1}\mathbf{P}^{-1}(\overline{\mathbf{b}} - \mathbf{u}), \tag{3.26}$$

and the associated covariance function

$$\operatorname{cov}(\delta\boldsymbol{\theta}^\star(t), \delta\boldsymbol{\theta}^\star(t')) = \boldsymbol{\Psi}(t)\,\mathbf{A}^{-1}\,\boldsymbol{\Psi}(t')^T. \tag{3.27}$$

To manipulate these expressions into a form that we can evaluate, some identities are employed. It can be shown using the Sherman-Morrison-Woodbury (SMW) identity (Sherman and Morrison, 1950; Woodbury, 1950) that

$$\mathbf{A}^{-1} \equiv \mathbf{P} - \mathbf{P}\boldsymbol{\Psi}^T\mathbf{H}^T\left(\mathbf{R} + \mathbf{H}\boldsymbol{\Psi}\mathbf{P}\boldsymbol{\Psi}^T\mathbf{H}^T\right)^{-1}\mathbf{H}\boldsymbol{\Psi}\mathbf{P}, \tag{3.28}$$

and

$$\mathbf{A}^{-1}\boldsymbol{\Psi}^T\mathbf{H}^T \equiv \mathbf{P}\boldsymbol{\Psi}^T\mathbf{H}^T\left(\mathbf{R} + \mathbf{H}\boldsymbol{\Psi}\mathbf{P}\boldsymbol{\Psi}^T\mathbf{H}^T\right)^{-1}\mathbf{R}. \tag{3.29}$$

Substituting these identities into (3.26)-(3.27), and reapplying the basis function relations (3.12)-(3.13) produces the GPGN update expressions

$$\begin{aligned}
\delta\boldsymbol{\theta}^\star(t) &= \boldsymbol{\Psi}(t)\,\mathbf{P}\boldsymbol{\Psi}^T\mathbf{H}^T\left(\mathbf{R} + \mathbf{H}\boldsymbol{\Psi}\mathbf{P}\boldsymbol{\Psi}^T\mathbf{H}^T\right)^{-1}\mathbf{R}\mathbf{R}^{-1}(\mathbf{z} - \mathbf{h}) \\
&\qquad - \boldsymbol{\Psi}(t)\left(\mathbf{P} - \mathbf{P}\boldsymbol{\Psi}^T\mathbf{H}^T\left(\mathbf{R} + \mathbf{H}\boldsymbol{\Psi}\mathbf{P}\boldsymbol{\Psi}^T\mathbf{H}^T\right)^{-1}\mathbf{H}\boldsymbol{\Psi}\mathbf{P}\right)\mathbf{P}^{-1}(\overline{\mathbf{p}} - \mathbf{u}) \\
&= \boldsymbol{\Psi}(t)\left(\mathbf{u} - \overline{\mathbf{p}}\right) + \boldsymbol{\Psi}(t)\,\mathbf{P}\boldsymbol{\Psi}^T\mathbf{H}^T\left(\mathbf{R} + \mathbf{H}\boldsymbol{\Psi}\mathbf{P}\boldsymbol{\Psi}^T\mathbf{H}^T\right)^{-1}\left(\mathbf{z} - \mathbf{h} - \mathbf{H}\boldsymbol{\Psi}\left(\mathbf{u} - \overline{\mathbf{p}}\right)\right) \\
&= \left(\boldsymbol{\eta}(t) - \overline{\boldsymbol{\theta}}(t)\right) + \boldsymbol{\mathcal{P}}(t)\,\mathbf{H}^T\left(\mathbf{R} + \mathbf{H}\boldsymbol{\mathcal{P}}\mathbf{H}^T\right)^{-1}\left(\mathbf{z} - \mathbf{h} - \mathbf{H}\left(\boldsymbol{\eta} - \overline{\boldsymbol{\theta}}\right)\right),
\end{aligned} \tag{3.30}$$

and

$$\begin{aligned}
\operatorname{cov}(\delta\boldsymbol{\theta}^\star(t), \delta\boldsymbol{\theta}^\star(t')) &= \boldsymbol{\Psi}(t)\left(\mathbf{P} - \mathbf{P}\boldsymbol{\Psi}^T\mathbf{H}^T\left(\mathbf{R} + \mathbf{H}\boldsymbol{\Psi}\mathbf{P}\boldsymbol{\Psi}^T\mathbf{H}^T\right)^{-1}\mathbf{H}\boldsymbol{\Psi}\mathbf{P}\right)\boldsymbol{\Psi}(t')^T \\
&= \boldsymbol{\Psi}(t)\,\mathbf{P}\boldsymbol{\Psi}(t')^T - \boldsymbol{\Psi}(t)\,\mathbf{P}\boldsymbol{\Psi}^T\mathbf{H}^T\left(\mathbf{R} + \mathbf{H}\boldsymbol{\Psi}\mathbf{P}\boldsymbol{\Psi}^T\mathbf{H}^T\right)^{-1}\mathbf{H}\boldsymbol{\Psi}\mathbf{P}\boldsymbol{\Psi}(t')^T \\
&= \boldsymbol{\mathcal{P}}(t,t') - \boldsymbol{\mathcal{P}}(t)\,\mathbf{H}^T\left(\mathbf{R} + \mathbf{H}\boldsymbol{\mathcal{P}}\mathbf{H}^T\right)^{-1}\mathbf{H}\boldsymbol{\mathcal{P}}(t')^T,
\end{aligned} \tag{3.31}$$

where

$$\begin{gathered}
\boldsymbol{\eta} := \begin{bmatrix}\boldsymbol{\mu} \\ \boldsymbol{\ell}\end{bmatrix}, \quad \boldsymbol{\mu} := \begin{bmatrix}\boldsymbol{\mu}(t_1) \\ \vdots \\ \boldsymbol{\mu}(t_N)\end{bmatrix}, \qquad \overline{\boldsymbol{\theta}} := \begin{bmatrix}\overline{\mathbf{x}} \\ \overline{\boldsymbol{\ell}}\end{bmatrix}, \quad \overline{\mathbf{x}} := \begin{bmatrix}\overline{\mathbf{x}}(t_1) \\ \vdots \\ \overline{\mathbf{x}}(t_N)\end{bmatrix}, \\
\boldsymbol{\mathcal{P}}(t) := \boldsymbol{\Psi}(t)\,\mathbf{P}\boldsymbol{\Psi}^T = \begin{bmatrix}\boldsymbol{\mathcal{K}}(t) & \mathbf{0} \\ \mathbf{0} & \mathbf{L}\end{bmatrix}, \quad \boldsymbol{\mathcal{K}}(t) := \boldsymbol{\Phi}(t)\,\mathbf{K}\boldsymbol{\Phi}^T = \begin{bmatrix}\boldsymbol{\mathcal{K}}(t,t_1) & \ldots & \boldsymbol{\mathcal{K}}(t,t_N)\end{bmatrix}, \\
\boldsymbol{\mathcal{P}} := \boldsymbol{\Psi}\mathbf{P}\boldsymbol{\Psi}^T = \begin{bmatrix}\boldsymbol{\mathcal{K}} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}\end{bmatrix}, \qquad \boldsymbol{\mathcal{K}} := \boldsymbol{\Phi}\mathbf{K}\boldsymbol{\Phi}^T = \begin{bmatrix}\boldsymbol{\mathcal{K}}(t_1,t_1) & \ldots & \boldsymbol{\mathcal{K}}(t_1,t_N) \\ \vdots & \ddots & \vdots \\ \boldsymbol{\mathcal{K}}(t_N,t_1) & \ldots & \boldsymbol{\mathcal{K}}(t_N,t_N)\end{bmatrix}.
\end{gathered} \tag{3.32}$$

The substitution of $\boldsymbol{\Psi}(t)\,\mathbf{P}\boldsymbol{\Psi}(t')^T$ by $\boldsymbol{\mathcal{P}}(t,t')$ makes use of the ‘kernel trick’, which recovers computational tractability. The $\boldsymbol{\mathcal{K}}$ and $\boldsymbol{\mathcal{K}}(t)$ matrices are constructed by simply evaluating the covariance function, $\boldsymbol{\mathcal{K}}(t,t')$, at the time of interest, t, and the measurement times, $t_1, \ldots, t_N$.

Further simplifications can be achieved if we consider the perturbations of the state at only the measurement times. That is, if we define $\delta\boldsymbol{\theta}^\star := \begin{bmatrix} \delta\mathbf{x}^{*^T} & \delta\boldsymbol{\ell}^{*^T} \end{bmatrix}^T$, where $\delta\mathbf{x}^\star := \begin{bmatrix} \delta\mathbf{x}^\star(t_1)^T & \ldots & \delta\mathbf{x}^\star(t_N)^T \end{bmatrix}^T$, we get

$$\delta\boldsymbol{\theta}^\star = \left(\boldsymbol{\eta} - \overline{\boldsymbol{\theta}}\right) + \boldsymbol{\mathcal{P}}\mathbf{H}^T\left(\mathbf{R} + \mathbf{H}\boldsymbol{\mathcal{P}}\mathbf{H}^T\right)^{-1}\left(\mathbf{z} - \mathbf{h} - \mathbf{H}\left(\boldsymbol{\eta} - \overline{\boldsymbol{\theta}}\right)\right), \tag{3.33}$$

$$\mathrm{cov}(\delta\boldsymbol{\theta}^\star, \delta\boldsymbol{\theta}^\star) = \boldsymbol{\mathcal{P}} - \boldsymbol{\mathcal{P}}\mathbf{H}^T\left(\mathbf{R} + \mathbf{H}\boldsymbol{\mathcal{P}}\mathbf{H}^T\right)^{-1}\mathbf{H}\boldsymbol{\mathcal{P}}, \tag{3.34}$$

and applying the SMW identity produces the update equations in information form:

$$\left(\boldsymbol{\mathcal{P}}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\right)\delta\boldsymbol{\theta}^\star = \boldsymbol{\mathcal{P}}^{-1}(\boldsymbol{\eta} - \overline{\boldsymbol{\theta}}) + \mathbf{H}^T\mathbf{R}^{-1}(\mathbf{z} - \mathbf{h}), \tag{3.35}$$

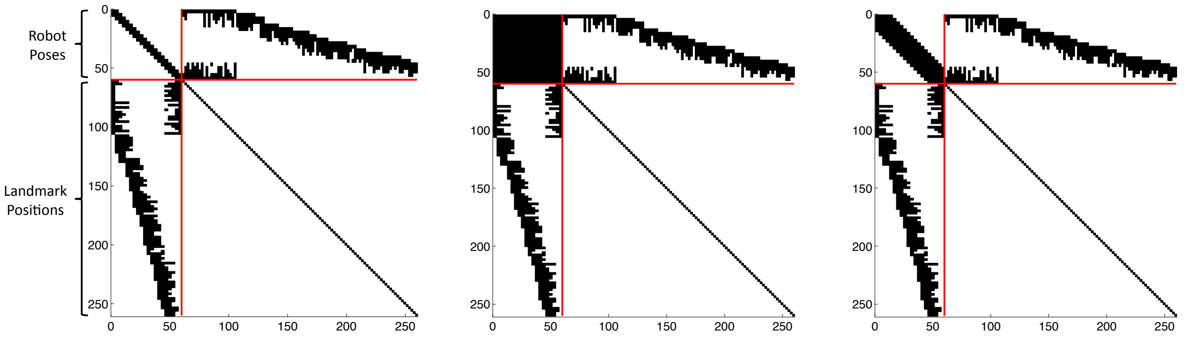
$$\mathrm{cov}(\delta\boldsymbol{\theta}^\star, \delta\boldsymbol{\theta}^\star)^{-1} = \boldsymbol{\mathcal{P}}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}. \tag{3.36}$$

These expressions have similar structure to the conventional discrete-time batch Gauss-Newton iterations (2.10)-(2.11). In fact, the estimate update expressions (3.35)-(3.36) simply have additional $\boldsymbol{\mathcal{P}}^{-1}$ and $\boldsymbol{\mathcal{P}}^{-1}(\boldsymbol{\eta} - \overline{\boldsymbol{\theta}})$ components, which serve as regularization terms (i.e., the GP prior serves as a smoothing factor for the solution).

The overall structure of the linear system of equations is not significantly modified by the inclusion of the regularization terms. This is depicted in Figure 3.2, where we compare the sparsity pattern of the inverse covariance matrix between the conventional discrete-time batch Gauss-Newton formulation with different choices of covariance functions for GPGN in the SLAM scenario. The lower-right block of the matrix remains block-diagonal, which allows for efficient solutions by applying the Schur complement. That is, if we let K be the number of poses and L be the number of landmarks, conditioning the robot poses on the landmark positions results in a complexity of $O(K^3 + K^2L)$. This is the same as the conventional discrete-time formulation.

The sparsity structure of the upper-left block of the matrix is related to the choice of GP prior. In Section 3.4, we utilize a dense covariance function, as depicted in Figure 3.2(b). However, as illustrated in Figure 3.2(c), sparsity can be recovered through the use of alternative covariance functions with compact support (Moreaux, 2008).

After solving for the optimal perturbation of the state at the measurement times using (3.35), it is applied as an additive update to the current state estimate by $\overline{\boldsymbol{\theta}} \leftarrow \overline{\boldsymbol{\theta}} + \delta\boldsymbol{\theta}^\star$. The system is then relinearized, and the process repeats until convergence.



(a) Discrete-time Gauss-Newton.

(b) GPGN with a dense covariance function.

(c) GPGN with a compactly supported covariance function.

Figure 3.2: Plots depicting the sparsity pattern of the inverse covariance matrix (3.36) for a sample 2D SLAM problem involving 20 robot poses and 100 landmarks. The red lines indicate the partitioning between the robot poses and landmark positions, and the landmarks are sorted in order of first observation. As can be seen, the additional regularization terms do not affect the lower-right block diagonal matrices, which allows the use of sparse solution techniques such as the application of the Schur complement. While the use of a dense covariance function results in complete fill-in for the upper-left block, some structure can be recovered through the use of compactly supported covariance functions instead.

3.3.4 Function-Space Derivation

The function-space derivation provides an alternative to the weight-space approach provided in the previous section. Both approaches achieve the same result (3.30)-(3.31), but the function-space derivation is considerably more concise.

We begin with our linearized measurement model

$$\mathbf{z}_i \approx \mathbf{h}_i(\overline{\boldsymbol{\theta}}(t_i)) + \mathbf{H}_i\, \delta\boldsymbol{\theta}(t_i) + \mathbf{n}_i, \tag{3.37}$$

and use it to obtain a Gaussian approximation. That is, the mean of the measurement function is

$$\begin{aligned} E\left[\mathbf{z}_i\right] &= E\left[\mathbf{h}_i(\overline{\boldsymbol{\theta}}(t_i)) + \mathbf{H}_i\, \delta\boldsymbol{\theta}(t_i) + \mathbf{n}_i\right] \\ &= \mathbf{h}_i(\overline{\boldsymbol{\theta}}(t_i)) + \mathbf{H}_i E\left[\delta\boldsymbol{\theta}(t_i)\right] \\ &= \mathbf{h}_i(\overline{\boldsymbol{\theta}}(t_i)) + \mathbf{H}_i \left(\boldsymbol{\eta}(t_i) - \overline{\boldsymbol{\theta}}(t_i)\right), \end{aligned} \tag{3.38}$$

and the covariance is

$$\begin{aligned} E\left[(\mathbf{z}_i - E[\mathbf{z}_i])(\mathbf{z}_i - E[\mathbf{z}_i])^T\right] &= E\Big[\left(\mathbf{n}_i - \mathbf{H}_i\left(\boldsymbol{\eta}(t_i) - \overline{\boldsymbol{\theta}}(t_i)\right)\right)\left(\mathbf{n}_i - \mathbf{H}_i\left(\boldsymbol{\eta}(t_i) - \overline{\boldsymbol{\theta}}(t_i)\right)\right)^T\Big] \\ &= E\left[\mathbf{n}_i\mathbf{n}_i^T\right] + \mathbf{H}_i\, E\Big[\left(\boldsymbol{\eta}(t_i) - \overline{\boldsymbol{\theta}}(t_i)\right)\left(\boldsymbol{\eta}(t_i) - \overline{\boldsymbol{\theta}}(t_i)\right)^T\Big]\mathbf{H}_i^T \\ &= \mathbf{R} + \mathbf{H}_i\boldsymbol{\mathcal{P}}(t_i, t_i)\,\mathbf{H}_i^T. \end{aligned} \tag{3.39}$$

These models can also be utilized to obtain the cross-covariance between the state and the measurements:

$$\begin{aligned} E\left[(\boldsymbol{\theta}(t) - E[\boldsymbol{\theta}(t)])(\mathbf{z}_i - E[\mathbf{z}_i])^T\right] &= E\Big[\left(\overline{\boldsymbol{\theta}}(t) + \delta\boldsymbol{\theta}(t) - \boldsymbol{\eta}(t)\right)\left(\mathbf{n}_i - \mathbf{H}_i\left(\boldsymbol{\eta}(t_i) - \overline{\boldsymbol{\theta}}(t_i)\right)\right)^T\Big] \\ &= E\left[\left(\overline{\boldsymbol{\theta}}(t) - \boldsymbol{\eta}(t)\right)\left(\overline{\boldsymbol{\theta}}(t_i) - \boldsymbol{\eta}(t_i)\right)^T\right]\mathbf{H}_i^T \\ &= \boldsymbol{\mathcal{P}}(t, t_i)\,\mathbf{H}_i^T. \end{aligned} \tag{3.40}$$

This Gaussian approximation is then used with the (3.20) and (3.32) definitions to express the joint distribution between the measurements and a single state perturbation as

$$\begin{bmatrix} \mathbf{z} \\ \delta\boldsymbol{\theta}(t) \end{bmatrix} \sim \mathcal{GP}\left(\begin{bmatrix} \mathbf{h} + \mathbf{H}(\boldsymbol{\eta} - \overline{\boldsymbol{\theta}}) \\ \boldsymbol{\eta}(t) - \overline{\boldsymbol{\theta}}(t) \end{bmatrix}, \begin{bmatrix} \mathbf{R} + \mathbf{H}\boldsymbol{\mathcal{P}}\mathbf{H}^T & \mathbf{H}\boldsymbol{\mathcal{P}}(t)^T \\ \boldsymbol{\mathcal{P}}(t)\mathbf{H}^T & \boldsymbol{\mathcal{P}}(t,t) \end{bmatrix}\right). \tag{3.41}$$

The optimal perturbation is then found by conditioning the state perturbation on the measurements, which produces

$$\begin{aligned} \delta\boldsymbol{\theta}^\star(t) \sim \mathcal{GP}\Big(&\left(\boldsymbol{\eta}(t) - \overline{\boldsymbol{\theta}}(t)\right) + \boldsymbol{\mathcal{P}}(t)\,\mathbf{H}^T\left(\mathbf{R} + \mathbf{H}\boldsymbol{\mathcal{P}}\mathbf{H}^T\right)^{-1}\left(\mathbf{z} - \mathbf{h} - \mathbf{H}\left(\boldsymbol{\eta} - \overline{\boldsymbol{\theta}}\right)\right), \\ &\boldsymbol{\mathcal{P}}(t,t') - \boldsymbol{\mathcal{P}}(t)\,\mathbf{H}^T\left(\mathbf{R} + \mathbf{H}\boldsymbol{\mathcal{P}}\mathbf{H}^T\right)^{-1}\mathbf{H}\boldsymbol{\mathcal{P}}(t')^T\Big). \end{aligned} \tag{3.42}$$

This is the same result as (3.30)-(3.31).

3.3.5 Linear Prediction Equation

After convergence, additional estimates for the continuous-time state component, $\mathbf{x}(t)$, at other times of interest can be obtained using (3.30)-(3.31). We use the term *prediction* for this task, since it is similar to the conventional use of GPs for regression. However, a more efficient expression for the mean function can be obtained through additional algebraic manipulation.

Starting with (3.30),

$$\delta\boldsymbol{\theta}^\star(t) = \boldsymbol{\eta}(t) - \overline{\boldsymbol{\theta}}(t) + \boldsymbol{\mathcal{P}}(t)\,\mathbf{H}^T\left(\mathbf{R} + \mathbf{H}\boldsymbol{\mathcal{P}}\mathbf{H}^T\right)^{-1}\left(\mathbf{z} - \mathbf{h} - \mathbf{H}\left(\boldsymbol{\eta} - \overline{\boldsymbol{\theta}}\right)\right), \tag{3.43}$$

we insert $\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{\mathcal{P}}$ to get

$$\delta\boldsymbol{\theta}^{\star}(t) = \boldsymbol{\eta}(t) - \overline{\boldsymbol{\theta}}(t) + \boldsymbol{\mathcal{P}}(t)\,\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{\mathcal{P}}\mathbf{H}^{T}\left(\mathbf{R} + \mathbf{H}\boldsymbol{\mathcal{P}}\mathbf{H}^{T}\right)^{-1}\left(\mathbf{z} - \mathbf{h} - \mathbf{H}\left(\boldsymbol{\eta} - \overline{\boldsymbol{\theta}}\right)\right), \tag{3.44}$$

and apply the SMW identity to result in

$$\delta\boldsymbol{\theta}^{\star}(t) = \boldsymbol{\eta}(t) - \overline{\boldsymbol{\theta}}(t) + \boldsymbol{\mathcal{P}}(t)\boldsymbol{\mathcal{P}}^{-1}\left(\boldsymbol{\mathcal{P}}^{-1} + \mathbf{H}^{T}\mathbf{R}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^{T}\mathbf{R}^{-1}\left(\mathbf{z} - \mathbf{h} - \mathbf{H}\left(\boldsymbol{\eta} - \overline{\boldsymbol{\theta}}\right)\right). \tag{3.45}$$

Next, we add and subtract $\boldsymbol{\mathcal{P}}^{-1}(\boldsymbol{\eta} - \overline{\boldsymbol{\theta}})$, and distribute the terms to obtain

$$\begin{aligned}\delta\boldsymbol{\theta}^{\star}(t) = \boldsymbol{\eta}(t) - \overline{\boldsymbol{\theta}}(t) + \boldsymbol{\mathcal{P}}(t)\boldsymbol{\mathcal{P}}^{-1}\left(\boldsymbol{\mathcal{P}}^{-1} + \mathbf{H}^{T}\mathbf{R}^{-1}\mathbf{H}\right)^{-1} \\ \times\left(\mathbf{H}^{T}\mathbf{R}^{-1}\left(\mathbf{z} - \mathbf{h}\right) - \mathbf{H}^{T}\mathbf{R}^{-1}\mathbf{H}\left(\boldsymbol{\eta} - \overline{\boldsymbol{\theta}}\right) + \boldsymbol{\mathcal{P}}^{-1}\left(\boldsymbol{\eta} - \overline{\boldsymbol{\theta}}\right) - \boldsymbol{\mathcal{P}}^{-1}\left(\boldsymbol{\eta} - \overline{\boldsymbol{\theta}}\right)\right),\end{aligned} \tag{3.46}$$

and simplify by grouping terms, which results in

$$\begin{aligned}\delta\boldsymbol{\theta}^{\star}(t) &= \boldsymbol{\eta}(t) - \overline{\boldsymbol{\theta}}(t) + \boldsymbol{\mathcal{P}}(t)\boldsymbol{\mathcal{P}}^{-1}\Big(-\left(\boldsymbol{\mathcal{P}}^{-1} + \mathbf{H}^{T}\mathbf{R}^{-1}\mathbf{H}\right)^{-1}\left(\boldsymbol{\mathcal{P}}^{-1} + \mathbf{H}^{T}\mathbf{R}^{-1}\mathbf{H}\right)\left(\boldsymbol{\eta} - \overline{\boldsymbol{\theta}}\right) \\ &\qquad + \underbrace{\left(\boldsymbol{\mathcal{P}}^{-1} + \mathbf{H}^{T}\mathbf{R}^{-1}\mathbf{H}\right)^{-1}\left(\boldsymbol{\mathcal{P}}^{-1}\left(\boldsymbol{\eta} - \overline{\boldsymbol{\theta}}\right) + \mathbf{H}^{T}\mathbf{R}^{-1}\left(\mathbf{z} - \mathbf{h}\right)\right)}_{\delta\boldsymbol{\theta}^{\star}}\Big) \\ &= \boldsymbol{\eta}(t) - \overline{\boldsymbol{\theta}}(t) - \boldsymbol{\mathcal{P}}(t)\boldsymbol{\mathcal{P}}^{-1}\left(\boldsymbol{\eta} - \overline{\boldsymbol{\theta}}\right) + \boldsymbol{\mathcal{P}}(t)\boldsymbol{\mathcal{P}}^{-1}\delta\boldsymbol{\theta}^{\star}.\end{aligned} \tag{3.47}$$

Since we are only interested in predicting values for the time-varying component of the state, we isolate and rearrange the expression to produce

$$\overline{\mathbf{x}}(t) + \delta\mathbf{x}^{\star}(t) = \boldsymbol{\mu}(t) - \boldsymbol{\mathcal{K}}(t)\boldsymbol{\mathcal{K}}^{-1}\left(\boldsymbol{\mu} - \overline{\mathbf{x}}\right) + \boldsymbol{\mathcal{K}}(t)\boldsymbol{\mathcal{K}}^{-1}\delta\mathbf{x}^{\star}. \tag{3.48}$$

Upon convergence of Gauss-Newton, $\delta\mathbf{x}^{\star}(t) = \delta\mathbf{x}^{\star} = \mathbf{0}$, which results in

$$\overline{\mathbf{x}}(t) = \boldsymbol{\mu}(t) - \boldsymbol{\mathcal{K}}(t)\boldsymbol{\mathcal{K}}^{-1}\left(\boldsymbol{\mu} - \overline{\mathbf{x}}\right). \tag{3.49}$$

This is an expression for predicting the value of the continuous-time state at other times of interest, which is composed of a linear combination of the converged state estimates at the measurement times. Furthermore, this expression also resembles the linear prediction equations for GP regression (Rasmussen and Williams, 2006). The covariance expression for a predicted state does not require any further manipulation, and remains as

$$\operatorname{cov}(\delta\mathbf{x}^{\star}(t), \delta\mathbf{x}^{\star}(t')) = \boldsymbol{\mathcal{K}}(t,t') - \boldsymbol{\mathcal{K}}(t)\,\mathbf{H}^{T}\left(\mathbf{R} + \mathbf{H}\boldsymbol{\mathcal{K}}\mathbf{H}^{T}\right)^{-1}\mathbf{H}\boldsymbol{\mathcal{K}}(t')^{T}. \tag{3.50}$$

3.3.6 Measurement Interpolation

The final step of the derivation involves utilizing the prediction equations to significantly reduce the computational resource requirements of GPGN. As noted earlier in Section 3.3.3.1, values for the state at each measurement time are required for linearization. This results in perturbations for the state to be estimated at all of the measurement times during each iteration, which can be prohibitive for large numbers of measurements. While the discrete-time formulation has the same requirements as this basic formulation of GPGN, we demonstrate another significant advantage of the continuous-time formulation by using interpolation to reduce the size of the estimated state.

The linear prediction equations from the previous section can be used to obtain the values of the state at some of the measurement times directly during the Gauss-Newton process. That is, if we substitute (3.48) during measurement linearization, we get

$$\begin{aligned}
\mathbf{z}_i &\approx \mathbf{h}_i(\overline{\boldsymbol{\theta}}(t_i)) + \mathbf{H}_i \,\delta\boldsymbol{\theta}(t_i) + \mathbf{n}_i \\
&= \mathbf{h}_i(\overline{\boldsymbol{\theta}}(t_i)) + \mathbf{H}_i \begin{bmatrix} \boldsymbol{\mu}(t_i) - \overline{\mathbf{x}}(t_i) - \boldsymbol{\mathcal{K}}(t_i)\,\boldsymbol{\mathcal{K}}^{-1}\,(\boldsymbol{\mu} - \overline{\mathbf{x}}) & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} + \mathbf{H}_i \begin{bmatrix} \boldsymbol{\mathcal{K}}(t_i)\,\boldsymbol{\mathcal{K}}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \delta\boldsymbol{\theta} + \mathbf{n}_i \\
&= \mathbf{h}_i(\overline{\boldsymbol{\theta}}(t_i)) + \mathbf{H}_i \begin{bmatrix} \boldsymbol{\mathcal{K}}(t_i)\,\boldsymbol{\mathcal{K}}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \delta\boldsymbol{\theta} + \mathbf{n}_i, \qquad (3.51)
\end{aligned}$$

where we utilized the standard assumption of convergence to evaluate $\overline{\mathbf{x}}(t_i)$ using (3.49). This expression is of a similar form as (3.19). As a result, we simply redefine $\mathbf{H}$ in (3.20) accordingly, and utilize the same equations as derived before.

We interpret this approach as *interpolation* because we are utilizing some state estimates to produce values at other times of interest. The mapping from the estimated to the interpolated states is provided by $\boldsymbol{\mathcal{K}}(t_i)\,\boldsymbol{\mathcal{K}}^{-1}$, which can be precomputed for each iteration. By estimating fewer states, we gain substantial computational savings. This method of approximation is similar to using less basis functions in the parametric approach, where we gain computational savings at the cost of representational fidelity.

However, the use of measurement interpolation also introduces additional operations in the form of interpolation matrix construction and dense fill-in, which must be balanced against the computational savings. If we let K be the number of estimated poses and P be the number of interpolated measurements, the complexity of these additional operations is $O(K^3 + K^2 P)$. While the optimization step is usually more costly, certain scenarios may arise where measurement interpolation results in an increase in computation time.

3.3.7 Summary

In summary, batch state estimation using the GPGN algorithm is performed as follows:

1. Divide the measurement times into two sets: the times at which the state values will be estimated, and the times at which the state values will be interpolated.
2. Initialize with an initial guess for the state at the estimated times, $\overline{\boldsymbol{\theta}}$.
3. Interpolate the state values for the remaining measurement times using (3.49).
4. Linearize and construct the (3.20) and (3.32) matrices.
5. Solve for the optimal perturbations at the estimated times, $\delta\boldsymbol{\theta}^\star$, using (3.35).
6. Apply the additive update to improve the state estimate, $\overline{\boldsymbol{\theta}} \leftarrow \overline{\boldsymbol{\theta}} + \delta\boldsymbol{\theta}^\star$.
7. Repeat steps 3-6 until convergence.
8. After convergence, compute the state and covariance of $\overline{\mathbf{x}}(t)$ at other timesteps of interest using (3.49) and (3.50).

The tradeoffs between this nonparametric approach and an explicit basis function parametrization (Furgale et al., 2012) are related to the time-varying state representation. In the case of parametric estimation, the computational complexity is proportional to the number of basis function coefficients. Similarly, the computational complexity of GPGN is proportional to the number of estimated states. In both cases, limiting the state size for computational savings comes at the cost of representational fidelity.

3.4 Experimental Validation

Since GPGN is a novel algorithm, experimental validation must be performed. We accomplish this by leveraging the robotics community's prior experience with the well-understood problem of 2D landmark-based SLAM. For comparison, GPGN was implemented alongside the conventional discrete-time batch Gauss-Newton approach (GN). Since GN utilizes all of the measurements in a discrete-time Maximum Likelihood formulation, it encompasses the traditional single-timestep smoothing approaches. The comparison is conducted using a combination of simulation and hardware experiments.

The intent of this section is not to convince the reader that 2D SLAM problems should be solved using GPGN; these are well addressed by the conventional methods. Rather, our intent is to illustrate the advantages of using the proposed algorithm without obfuscation by complex experimental configurations. While we demonstrate improved estimation accuracy with a small increase in computational cost, we remind the reader that the key advantage of the continuous-time formulation is the ability to handle measurement scarcity and asynchronicity. These issues cannot be handled by the discrete-time formulation in a computationally tractable manner.

We begin with a brief overview of the experimental scenario, and then provide some implementation details for the GPGN and GN algorithms. This is followed by the simulation results, and concluded with the hardware experiment.

3.4.1 Overview

In this scenario, we considered a mobile robot equipped with a laser rangefinder driving in a planar environment. Two measurement types were available: instantaneous linear and angular velocities through wheel odometry, and range and bearing measurements to point landmarks in the scene using the laser rangefinder. Known data association was provided, and the maximum observation range was artificially limited to 2m. The odometry and landmark measurements were provided to the estimators at a rate of 1Hz, and after estimator convergence, additional estimates were computed at a rate of 10Hz to evaluate performance. To emulate the ability of GPGN to produce additional estimates between measurement times, linear interpolation was employed for the GN estimates.

This experimental scenario was carefully designed to allow for easy comparison between the two algorithms. Both odometry and landmark measurements were synchronized, and known data association was provided to allow for unit testing of the estimators themselves. In addition, we demonstrate the advantage provided by the underlying process model in the continuous-time formulation by considering the landmark-only scenario for the hardware experiment data.

3.4.2 GPGN Implementation

For GPGN, we defined our time-varying state vector to be

$$\mathbf{x}(t) := \begin{bmatrix} x(t) & y(t) & \theta(t) \end{bmatrix}^T, \tag{3.52}$$

the robot's pose at time t, and the time-invariant component to be the landmark locations, $\boldsymbol{\ell}$. We utilized a continuous white noise acceleration (CWNA) model (Bar-Shalom et al., 2002) for the time-varying state, where

$$\ddot{\mathbf{x}}(t) \sim \mathcal{GP}\left(\mathbf{0}, \mathbf{W}\,\delta(t-t')\right), \tag{3.53}$$

$\mathbf{W}$ is the power spectral density matrix, and $\delta(\cdot)$ is the Dirac delta function. This model can be interpreted as a constant velocity assumption with perturbative noise. While this model does not utilize any additional knowledge about the hardware configuration, it is simple, and it provides a good starting point with some physical intuition. Possible models that may be investigated in the future are discussed in Section 3.5. As shown in Appendix A, integrating the mean and covariance functions twice produced the GP for $\mathbf{x}(t)$, where

$$\boldsymbol{\mu}(t) = \mathbf{x}(0), \tag{3.54}$$

$$\boldsymbol{\mathcal{K}}(t,t') = \mathbf{W}\left(\frac{\min(t,t')^2\max(t,t')}{2} - \frac{\min(t,t')^3}{6}\right). \tag{3.55}$$

Along with this process model, the instantaneous linear and angular velocity odometry measurements were modelled as

$$\mathbf{v}_k := \begin{bmatrix} \cos\theta(t_k) & \sin\theta(t_k) & 0 \\ -\sin\theta(t_k) & \cos\theta(t_k) & 0 \\ 0 & 0 & 1 \end{bmatrix} \dot{\mathbf{x}}(t) + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0},\mathbf{Q}), \tag{3.56}$$

where we allowed for a small amount of lateral wheel slip, and $\mathbf{w}_k$ was corruptive zero-mean Gaussian noise with a covariance of $\mathbf{Q}$. This minimal lateral wheel slip model implicitly accounts for the nonholonomic nature of the robot platform. The $\dot{\mathbf{x}}(t)$ values required for linearization were obtained using the linear prediction equation (3.49), and the odometry measurements were incorporated using measurement interpolation (3.51). The approach for predicting derivative state values and incorporating derivative-type measurements into GPGN is detailed in Appendix A.2.

Finally, the laser rangefinder measurements to landmark j were modelled as

$$\mathbf{z}_i := \begin{bmatrix} \left((x_j - x(t_i) - d\cos\theta(t_i))^2 + (y_j - y(t_i) - d\sin\theta(t_i))^2\right)^{1/2} \\ \operatorname{atan2}\left(y_j - y(t_i) - d\sin\theta(t_i), x_j - x(t_i) - d\cos\theta(t_i)\right) - \theta(t_i) \end{bmatrix} + \mathbf{n}_{i,l},$$

$$\mathbf{n}_{i,l} \sim \mathcal{N}(\mathbf{0},\mathbf{R}), \tag{3.57}$$

where (x_j, y_j) was the position of landmark j, d was the offset between the robot and sensor centres, and $\mathbf{n}_{i,l}$ was the corruptive Gaussian noise with a covariance of $\mathbf{R}$. An illustrated definition of these state parameters is provided in Figure 3.3.

For all three system models, the noise parameters were determined during a prior characterization process that utilized ground truth data and fitted Gaussian distributions to the computed errors from a different run than the testing data. We can consider this Gaussian fitting as hyperparameter training for our GP process model.

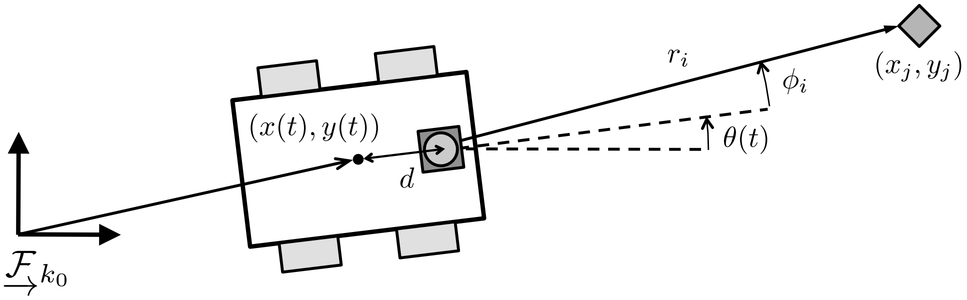


Figure 3.3: An illustration of the states involved in the experimental trials. The robot state was composed of its position, $(x(t), y(t))$, and orientation, $\theta(t)$, and the laser rangefinder measurements to landmark j contained the range, r_i, and bearing, ϕ_i. An additional offset, d, was included to account for the difference between the robot and sensor centres.

3.4.3 Discrete-Time Batch Gauss-Newton Implementation

Similarly, for GN, we defined our state vector to be the robot pose at timestep k,

$$\mathbf{x}_k := \begin{bmatrix} x_k & y_k & \theta_k \end{bmatrix}^T, \tag{3.58}$$

and the landmark locations as $\boldsymbol{\ell}$.

The discrete-time odometry measurement model was obtained by performing an explicit Euler step of the continuous-time model, which resulted in

$$\mathbf{v}_k := \frac{1}{T} \begin{bmatrix} \cos\theta_k & \sin\theta_k & 0 \\ -\sin\theta_k & \cos\theta_k & 0 \\ 0 & 0 & 1 \end{bmatrix} (\mathbf{x}_{k+1} - \mathbf{x}_k) + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}), \tag{3.59}$$

where T was the sampling period. The laser measurement model remained the same as the GPGN case (3.57) with discrete poses, $\mathbf{x}_k$, in place of the $\mathbf{x}(t)$ values. The measurement data and noise parameters were kept identical for both algorithms.

3.4.4 Simulation Results

To facilitate statistical analysis, we conducted 1000 simulated trials of a 2D SLAM problem. Figure 3.4 depicts the layout we considered, where the robot travelled two loops in a figure-eight pattern, and observed small clusters of landmarks throughout its traverse. With the 1Hz measurement rate, each trial consisted of 300 robot poses and 100 landmarks obtained over a 5 minute time window.

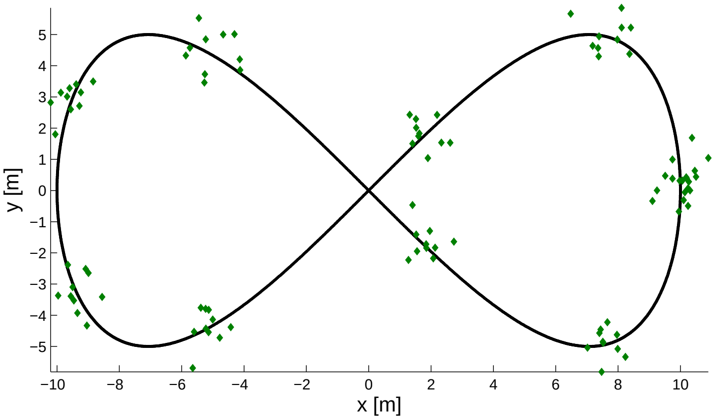


Figure 3.4: A typical layout for a simulation trial consisting of 300 robot poses and 100 landmarks over a 5 minute traverse. The ground truth robot path is indicated by the black line, and the landmark positions by the green diamonds.

The average root mean squared (RMS) estimation errors over the 1000 trials were computed for quantitative comparison. We evaluate both mapping and localization performance by computing the errors under the maximally consistent alignment between the estimated and ground truth maps. These results are presented in Table 3.1. Compared to GN, GPGN achieved improvements in accuracy of 13.1% in translation, 14.5% in orientation, and 10.4% in landmark position.

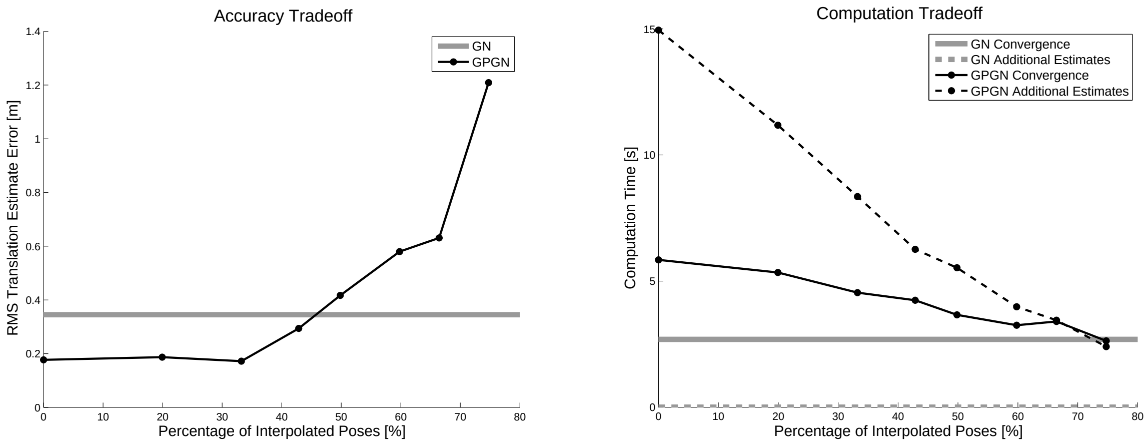
However, these improvements in estimation accuracy also came with an increase in computation time[1]. Though GPGN required three times the computation time for convergence, the total amount of time is still significantly less than the experiment time window of 5 minutes. Furthermore, the additional estimates would only be needed during post-processing, and can be computed in an offline manner.

As introduced in Section 3.3.6, we can reduce the computational requirements by interpolating some of the robot poses. To give a sense of the tradeoffs in accuracy and

[1] Timing information recorded on a MacBook Pro with a 2.66GHz Core 2 Duo and 4GB of 1067MHz DDR3 RAM in Matlab, utilizing both cores.

Table 3.1: Average RMS estimate errors and computation time for the simulated trials.

	Translation Error [m]	Orientation Error [°]	Landmark Position Error [m]	Convergence Time [s]	Additional Estimate Interpolation Time [s]
GN	0.245	3.11	0.251	2.5	0.003
GPGN	0.213	2.66	0.225	7.3	15



(a) RMS translation error.

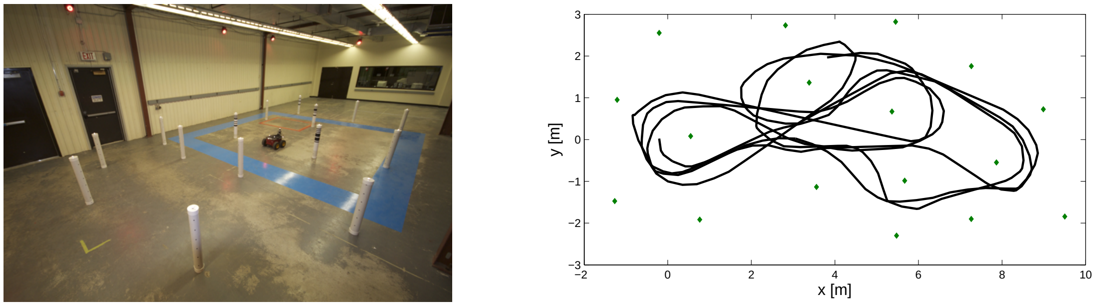
(b) Computation time for convergence and obtaining the additional pose estimates.

Figure 3.5: Plots depicting the tradeoffs in GPGN estimation accuracy and computation time with respect to the percentage of interpolated poses for a single simulation trial consisting of 300 robot poses and 100 landmarks. For comparison, the GN values are also depicted by the horizontal lines. As can be seen, the number of estimated poses could be reduced without significant detriment to the accuracy. In other words, only a subset of the poses were required to represent the robot trajectory.

computation, we varied the interpolation percentage for one of the simulation trials. The results are depicted in Figure 3.5, where the RMS translation error and computation times for GPGN are plotted with respect to the percentage of interpolated poses. As can be seen, the number of estimated poses could be moderately reduced without significantly affecting the estimation accuracy. This is likely due to the fact that the figure-eight trajectory can be well-represented by a only subset of the robot poses.

3.4.5 Hardware Experiment

The experimental setup consisted of a mobile robot driving in an indoor, planar environment, amongst a forest of plastic tubes, depicted in Figure 3.6(a). These tubes served as landmarks for our localization problem, and ground truth data was obtained using a Vicon motion capture system that tracked retroreflective markers placed on both the robot and the plastic tubes. Due to the limited size of the workspace, only 17 landmarks were available in this experiment. As depicted in Figure 3.6(b), the robot traverse over 5 minutes was sufficiently varied for our evaluation purposes.



(a) The hardware consisted of a P3-AT mobile robot, a Hokuyo URG-04LX-UG01 laser rangefinder, plastic tubes, and a Vicon motion tracking system.

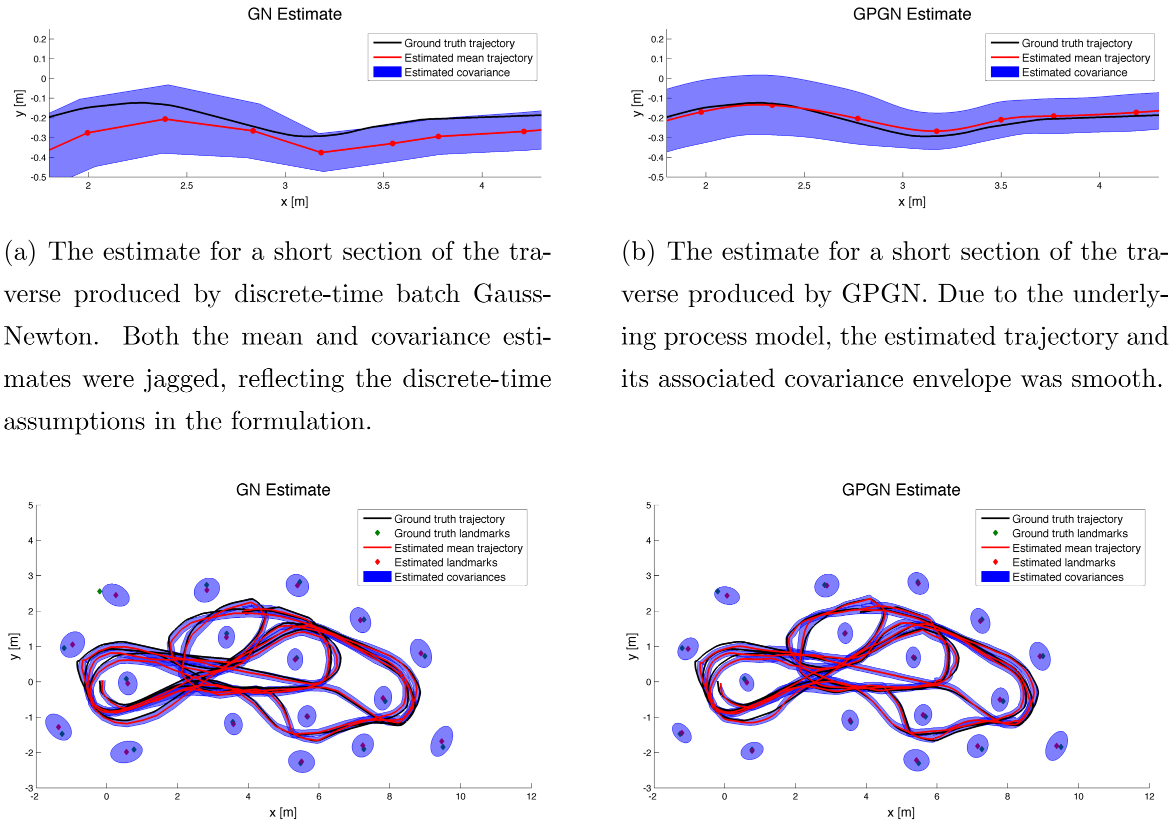
(b) An overhead plot of the experimental data obtained during the 5 minute traverse. The ground truth robot path is indicated by the black line, and the landmarks by the green diamonds.

Figure 3.6: The hardware configuration for the 2D SLAM experiment utilized for validation of the GPGN algorithm.

To highlight the differences between the two approaches, we begin by focusing on a small section of the traverse. In this section, the robot travelled from left to right, with small corrective turns along the way. The results produced by the two estimators, depicted in Figures 3.7(a)-3.7(b), were quite different. The GN estimate in Figure 3.7(a) was quite jagged, modelling the trajectory as a series of discrete poses. This reflects the formulation of the discrete-time approach. On the other hand, Figure 3.7(b) shows that the underlying process model provided a much smoother estimate for GPGN, resulting in a better match to the actual robot trajectory. These observations hold for both the mean and covariance estimates.

These small differences between the two algorithms compound into large estimation errors for the whole traverse, depicted in Figures 3.7(c)-3.7(d). The jagged sections in the GN estimate result in reduced performance at the edges, including a landmark position estimate that exceeded the 3σ covariance error bounds. Meanwhile, GPGN was able to produce smooth estimates for both the mean and covariance, which matched the ground truth data more accurately.

Quantitative comparison of the relative performance is presented in Table 3.2, where we see that GPGN achieved improvements in accuracy of 23.7% in translation, 23.5% in orientation, and 36.2% in landmark position. Much like the simulation trials, these improvements in estimation accuracy also came with an increase in computation time. Once again, though GPGN required five times the computation time for convergence, the total amount of time was still less than the 5 minutes of experimental data.



(a) The estimate for a short section of the traverse produced by discrete-time batch Gauss-Newton. Both the mean and covariance estimates were jagged, reflecting the discrete-time assumptions in the formulation.

(b) The estimate for a short section of the traverse produced by GPGN. Due to the underlying process model, the estimated trajectory and its associated covariance envelope was smooth.

(c) The estimate for the whole 5 minutes produced by discrete-time batch Gauss-Newton. Though the performance was quite accurate for the majority of the traverse, jagged estimates were still present, resulting in reduced performance at the edges.

(d) The estimate for the whole 5 minutes produced by GPGN. The smooth estimated trajectory and its associated covariance envelope accurately matched the ground truth path.

Figure 3.7: Plots depicting the results of the two estimators for both a short section and the entire 5 minute robot traverse. The ground truth trajectory is indicated by the black line, and the estimated rover positions, along with their associated 3σ lateral covariance envelopes, are indicated by the red line and the blue shading, respectively. The lateral covariance envelopes were determined by projecting the estimated (x, y) covariances onto the lateral dimension defined by the mean orientation estimate.

Finally, we demonstrate the key advantage of using the continuous-time approach by discarding the wheel odometry and only utilizing the landmark measurements. As depicted in Figure 3.8, 25% of the poses have fewer than two landmark measurements. While GN is unable to compute an estimate at those locations, the issues of measurement scarcity are avoided by GPGN due to the presence of an underlying process model.

Table 3.2: RMS estimate errors and computation time for the hardware experiment.

	Translation Error [m]	Orientation Error [°]	Landmark Position Error [m]	Convergence Time [s]	Additional Estimate Interpolation Time [s]
GN	0.114	4.38	0.177	1.6	0.002
GPGN	0.087	3.35	0.113	7.4	15
LM-GPGN	0.077	3.01	0.071	31	15

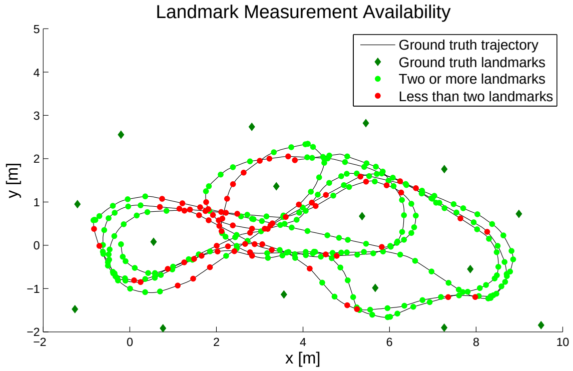
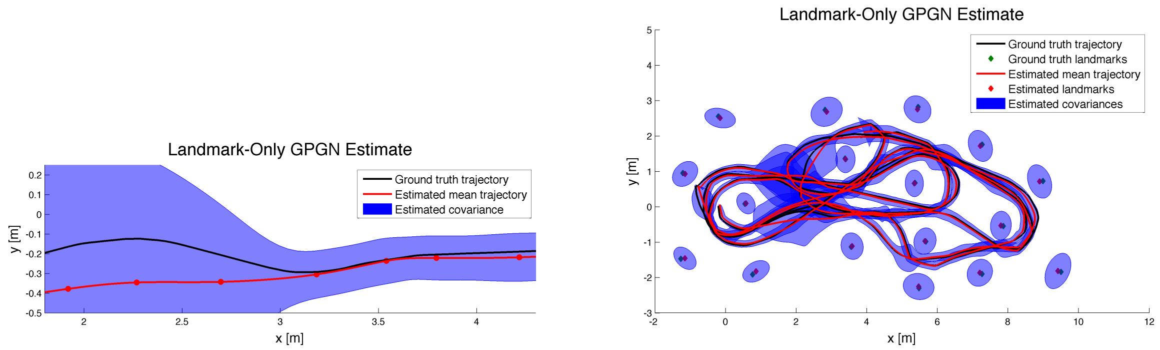


Figure 3.8: The number of landmark measurements visible at each robot pose. The green dots indicate poses that observe two or more landmarks, and the red dots indicate poses that do not. Since a minimum of two landmark measurements are required to produce a 2D pose estimate, the discrete-time batch Gauss-Newton formulation is unable to address this landmark-only scenario.

Figure 3.9 depicts the results of GPGN estimation using only the landmark measurements (LM-GPGN) for both the small section of the traverse, as well as the overall trajectory. In Figure 3.9(a), we see a much larger lateral covariance envelope, which can be attributed to the reduced number of measurements, and the omission of the minimal lateral wheel slip constraint provided by the odometry measurements. The overall traverse depicted in Figure 3.9(b) is similar to the plots in Figures 3.7(c)-3.7(d), except for the larger covariance envelopes in the regions of measurement scarcity. Since an insufficient amount of information is available at these timesteps, the uncertainty is appropriately higher in these regions. Finally, the quantitative results attributed to LM-GPGN in Table 3.2 demonstrate that a similar level of accuracy is achieved. The slight improvement in estimator accuracy may be attributed to an inaccurate wheel odometry model, and the large increase in estimation time was due to the omission of the minimal lateral wheel slip constraint. This resulted in a larger search space during optimization, which required many more iterations for convergence.



(a) The estimate for the short section of the robot traverse depicted in Figures 3.7(a)-3.7(b). The smooth growth of the covariance envelope reflects the measurement scarcity, and the omission of the odometry measurements.

(b) The estimate for the overall traverse. In addition to being more accurate than the plots in Figures 3.7(c)-3.7(d), it can be seen that the covariance envelope expands smoothly and accurately in the regions with measurement scarcity.

Figure 3.9: Plots depicting the results of landmark-only GPGN estimator. The ground truth trajectory and landmark locations are indicated by the black line and green diamonds, respectively. The estimated robot positions and landmark estimates are indicated in red, and the associated 3σ covariance envelopes are represented by the blue shading.

3.5 Conclusion

In conclusion, we have introduced GPGN, a novel continuous-time algorithm for nonparametric, nonlinear, batch state estimation. This work adapts concepts from GP regression, and applies it to the robotics domain. GPGN is significantly different from the previous uses of GPs in robotics. In particular, we do not use GPs for output modelling; rather, we use GPs to model the underlying state. As a result, GPGN is able to address conventional robotic state estimation scenarios.

While we deliberately avoided presenting a complex experimental scenario for validation of GPGN, the experimental results demonstrated the tradeoffs of GPGN compared with the conventional discrete-time batch Gauss-Newton approach. These advantages include the production of smooth mean and covariance estimates that better reflect the physical reality of the scenario, and the ability to fall back on the underlying continuous-time process model in cases of measurement scarcity. This was demonstrated by the ability of GPGN to address the landmark-only SLAM scenario. While the improvements in modelling accuracy come with increased computational cost, our experimental results demonstrate that the increase is reasonable for practical problems.

In our experiments, we employed a simple GP model based on the assumption of constant velocity with perturbative noise. While we demonstrated improvements in estimation accuracy, this simple model did not make use of any additional knowledge about the robot platform. This may be addressed by incorporating a parametric model into the mean function (Ko and Fox, 2009; Guizilini and Ramos, 2012). In effect, the expected motion of the platform is represented by the mean, and the remaining unmodelled disturbances are captured by the covariance function. This approach can also employ the control inputs, if available.

However, our derivation of GPGN does not make any assumptions on the form of the GP. As a result, we are able to utilize any GP in this algorithm, whether dense, sparse, stationary, or heteroscedastic. Furthermore, a large amount of literature is also available on scaling GPs to large systems, including approximation using sparse methods, and compactly supported covariance functions (Rasmussen and Williams, 2006). Online operation may be addressed by utilizing a sliding window formulation (Sibley et al., 2010a), or by extending the Incremental Smoothing and Mapping (iSAM) (Kaess et al., 2008, 2012) factorizations to GPGN, which will provide the full batch solution at a fraction of the computational cost. These factorizations have been utilized successfully for online GP regression (Ranganathan et al., 2011), and we expect a similar benefit for continuous-time state estimation. Many of the variations on the basic framework of GP regression should be transferable to the robotics domain.

As will be shown in Chapter 6, this novel continuous-time approach to state estimation enables the computation of VO estimates from motion-distorted laser intensity imagery.

Chapter 4

State Estimation in Three Dimensions

4.1 Introduction

In the previous chapters, we derived two batch state estimation algorithms using linear algebra. While experimental validation was demonstrated for a 2D SLAM problem, some complexities arise when attempting to apply these algorithms to the 3D domain. The primary challenge in 3D estimation stems from the nature of rotations, which conflicts with the vector space assumptions that underlie the derivations.

In particular, we assumed that the state was represented by a vector. This permitted additive perturbations for the Gauss-Newton iterations, and provided simple Jacobian matrices during measurement model linearization. Issues can arise in all of these areas when conducting 3D estimation, and as a result, are addressed in this chapter.

As mentioned in Chapter 2, the key distinction between many estimation algorithms is the state parametrization. A variety of options are available for rotation representation (Stuelpnagel, 1964), each with a different set of tradeoffs. Vector representations such as Euler angles and quaternions provide some vector space notions, but suffer from singularities or require additional constraints (Shuster, 1993). Alternatively, rotation matrices uniquely define the entire space, but are members of a *noncommutative group*, instead of a vector space. However, a local vector space may be constructed for small rotations, and constraint-sensitive perturbations may be employed to maintain membership within the group (Barfoot et al., 2011). These properties fit well within the Gauss-Newton framework.

We utilize this approach in our work, but consider a compact representation for the rover poses, where the translation and rotation components are combined into transformation matrices (Furgale and Barfoot, 2013). Since this matrix parametrization violates the vector state assumption, we redefine the perturbation and linearization approach for this parametrization. While there are numerous other useful properties identified from previous applications in manipulator robotics (Murray et al., 1994) and computer vision (Hartley and Zisserman, 2000), we will only provide the core concepts for implementation.

During the course of this thesis, an alternative parametrization of translation vectors and rotation matrices was employed for poses in the mapping framework. However, these representations are equivalent, and as a result, we take the opportunity to unify the notation in this thesis document. The following sections provide a brief overview of our state parametrization, including useful identities and properties. This is followed by our state perturbation approach, and derivations of common linearized measurement error models. Finally, we conclude with a summary of the key modifications required for extending the batch state estimation algorithms to the 3D domain.

4.2 State Parametrization

We represent rover poses using transformation matrices, and feature vectors by homogeneous vectors (Murray et al., 1994; Hartley and Zisserman, 2000). These quantities are expressed with respect to a single estimate reference frame, $\underrightarrow{\mathcal{F}}_{k_0}$. That is, if we define $\boldsymbol{\rho}_{k_0}^{k,k_0}$ to be the 3×1 translation vector from $\underrightarrow{\mathcal{F}}_{k_0}$ to $\underrightarrow{\mathcal{F}}_{k}$, expressed in $\underrightarrow{\mathcal{F}}_{k_0}$, and $\mathbf{C}_{k,k_0}$ to be the 3×3 rotation matrix from $\underrightarrow{\mathcal{F}}_{k_0}$ to $\underrightarrow{\mathcal{F}}_{k}$, we can define a 4×4 transformation matrix as

$$\boldsymbol{T}_{k,k_0} := \begin{bmatrix} \mathbf{C}_{k,k_0} & -\mathbf{C}_{k,k_0}\boldsymbol{\rho}_{k_0}^{k,k_0} \\ \mathbf{0}^T & 1 \end{bmatrix}. \tag{4.1}$$

If we also define $\mathbf{p}_{k_0}^{j,k_0}$ to be the 3×1 position of feature j with respect to $\underrightarrow{\mathcal{F}}_{k_0}$, expressed in $\underrightarrow{\mathcal{F}}_{k_0}$, we can represent feature positions as 4×1 vectors defined in homogeneous coordinates, where

$$\boldsymbol{p}_{k_0}^{j,k_0} := \begin{bmatrix} \mathbf{p}_{k_0}^{j,k_0} \\ 1 \end{bmatrix}. \tag{4.2}$$

These definitions allow for a single multiplication operation to transform a point expressed in the estimate frame, $\underrightarrow{\mathcal{F}}_{k_0}$, to a local frame, $\underrightarrow{\mathcal{F}}_{k}$, by

$$\boldsymbol{p}_{k}^{j,k} = \boldsymbol{T}_{k,k_0}\boldsymbol{p}_{k_0}^{j,k_0}. \tag{4.3}$$

4.2.1 Useful Properties

A key property of transformation matrices is that they belong to the matrix Lie group, $SE(3)$, which means that there is a corresponding Lie algebra, $\mathfrak{se}(3)$ (Stillwell, 2008). These representations are related by the (matrix) exponential map (Murray et al., 1994; Sastry, 1999; Stillwell, 2008). That is, we can define $\boldsymbol{\pi}$, the 6×1 vector representation of the pose corresponding to the transformation matrix $\boldsymbol{T}$, as

$$\boldsymbol{T} =: e^{-\boldsymbol{\pi}^{\boxplus}}, \quad \boldsymbol{\pi} := \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix}, \tag{4.4}$$

where $\boldsymbol{\rho}$ is a 3×1 vector corresponding to the translation, $\boldsymbol{\phi}$ is a 3×1 vector corresponding to the rotation, $(\cdot)^{\boxplus}$ is the linear matrix operator that transforms a 6×1 vector into a 4×4 matrix according to

$$\boldsymbol{\pi}^{\boxplus} = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix}^{\boxplus} := \begin{bmatrix} \boldsymbol{\phi}^{\times} & -\boldsymbol{\rho} \\ \mathbf{0}^T & 0 \end{bmatrix}, \tag{4.5}$$

and $(\cdot)^{\times}$ is the linear skew-symmetric matrix operator (Hughes, 1986), defined as

$$\boldsymbol{\phi}^{\times} := \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}^{\times} := \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix}. \tag{4.6}$$

Our specific definition[1] of $(\cdot)^{\boxplus}$ originally appeared in Furgale (2011), but similar operators can be found in Murray et al. (1994) and Sastry (1999). We employ this mapping to define a constraint-sensitive state perturbation in the following section.

The mapping from the vector parametrization to a transformation matrix is provided by the exponential map, $\boldsymbol{T} = e^{-\boldsymbol{\pi}^{\boxplus}}$, which is implemented efficiently by partitioning $\boldsymbol{\pi}$ into its $\boldsymbol{\rho}$ and $\boldsymbol{\phi}$ components from (4.4), computing the matrix $\mathbf{S}$, where

$$\begin{aligned} \mathbf{S}(\phi) &:= \frac{\sin\phi}{\phi}\mathbf{1} + \left(1 - \frac{\sin\phi}{\phi}\right)\mathbf{a}\mathbf{a}^T + \frac{\cos\phi - 1}{\phi}\mathbf{a}^{\times}, \\ \mathbf{a} &:= \frac{\boldsymbol{\phi}}{\phi}, \\ \phi &:= \|\boldsymbol{\phi}\|_2, \end{aligned} \tag{4.7}$$

[1] The negative sign in our definition of $(\cdot)^{\boxplus}$ differs from some others in the literature. We selected this convention to be consistent with the definition of rotations in Hughes (1986). This matrix operator also appears as a $\hat{(\cdot)}$ in the Lie algebra literature (Murray et al., 1994).

and determining the rotation matrix $\mathbf{C}$ using

$$\mathbf{C} := \mathbf{1} - \boldsymbol{\phi}^{\times}\mathbf{S}. \tag{4.8}$$

After obtaining $\mathbf{C}$, $\mathbf{S}$, and $\boldsymbol{\rho}$, we can then construct the transformation matrix $\boldsymbol{T}$, from

$$\boldsymbol{T}(\boldsymbol{\pi}) =: \begin{bmatrix} \mathbf{C} & \mathbf{r} \\ \mathbf{0}^T & 1 \end{bmatrix} := \begin{bmatrix} \mathbf{C} & \mathbf{S}\boldsymbol{\rho} \\ \mathbf{0}^T & 1 \end{bmatrix}. \tag{4.9}$$

The inverse mapping is provided by the logarithmic map, $\ln \boldsymbol{T} = -\boldsymbol{\pi}^{\boxplus}$. This is implemented by first extracting the $\mathbf{C}$ and $\mathbf{r}$ components from $\boldsymbol{T}$, and obtaining $\boldsymbol{\phi}$ using

$$\begin{aligned} \boldsymbol{\phi}(\mathbf{C}) &= \phi\mathbf{a}, \\ \mathbf{a} &:= \frac{1}{2\sin\phi}\begin{bmatrix} C_{32} - C_{23} \\ C_{13} - C_{31} \\ C_{21} - C_{12} \end{bmatrix}, \\ \phi &:= \cos^{-1}\left(\frac{\operatorname{trace}(\mathbf{C}) - 1}{2}\right), \end{aligned} \tag{4.10}$$

where C_{ij} indicates the ith row and jth column entry of the $\mathbf{C}$ matrix. It should be noted that the sign of $\boldsymbol{\phi}$ may need to be verified by inserting ϕ and $\mathbf{a}$ back into (4.8). Using $\boldsymbol{\phi}$, we compute $\mathbf{S}$ from (4.7), and determine $\boldsymbol{\rho}$ with

$$\boldsymbol{\rho} = \mathbf{S}^{-1}\mathbf{r}. \tag{4.11}$$

The $\boldsymbol{\rho}$ and $\boldsymbol{\phi}$ are then combined into the vector parametrization, $\boldsymbol{\pi}$, using (4.4).

There are also a few additional useful identities involving the $(\cdot)^{\boxplus}$ matrix operator that we employ during linearization of the measurement error models. These are

$$\left(\mathbf{1} - \mathbf{w}^{\boxplus}\right)^{-1} \approx \mathbf{1} + \mathbf{w}^{\boxplus}, \tag{4.12}$$

$$\mathbf{w}^{\boxplus}\boldsymbol{y} = -\boldsymbol{y}^{\boxminus}\mathbf{w}, \tag{4.13}$$

$$\left(\boldsymbol{T}^{\boxdot}\mathbf{w}\right)^{\boxplus} = \boldsymbol{T}\mathbf{w}^{\boxplus}\boldsymbol{T}^{-1}, \tag{4.14}$$

where $\mathbf{w}$ is a 6×1 vector, $\boldsymbol{y}$ is a 4×1 homogeneous vector, and the two matrix operators, $(\cdot)^{\boxminus}$ and $(\cdot)^{\boxdot}$, are defined as

$$\boldsymbol{y}^{\boxminus} := \begin{bmatrix} \mathbf{e} \\ n \end{bmatrix}^{\boxminus} := \begin{bmatrix} n\mathbf{1} & \mathbf{e}^{\times} \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix}, \tag{4.15}$$

$$\boldsymbol{T}^{\boxdot} := \begin{bmatrix} \mathbf{C} & \mathbf{r} \\ \mathbf{0}^T & 1 \end{bmatrix}^{\boxdot} := \begin{bmatrix} \mathbf{C} & -\mathbf{r}^{\times}\mathbf{C} \\ \mathbf{0}^T & \mathbf{C} \end{bmatrix}. \tag{4.16}$$

4.3 Perturbations and Uncertainty

With our state defined as a collection of transformation matrices and homogeneous feature vectors, we also require a definition for perturbations and uncertainty as it relates to this parametrization. These definitions are necessary to maintain the probabilistic formulation, as well as to establish a form for the optimal state perturbations computed during each iteration of the Gauss-Newton algorithm. Though our state is not a vector, we note the fact that the iterative update equations only involve the perturbations. As a result, we can still employ the Gauss-Newton algorithm for estimation if we are able to provide a local vector space for small perturbations.

For transformation matrices, we use an injection of noise onto the group, $SE(3)$, by employing the exponential map (Furgale and Barfoot, 2013). That is,

$$\boldsymbol{T} = e^{-\delta\boldsymbol{\pi}^{\boxplus}}\overline{\boldsymbol{T}}, \tag{4.17}$$

where $\delta\boldsymbol{\pi}$ is a 6×1 noise vector, and $\overline{\boldsymbol{T}}$ is a noise-free pose. By injecting the uncertainty through the exponential map, we ensure that the noisy random variable $\boldsymbol{T}$ is still within $SE(3)$. As a result, the uncertainty in the matrix $\boldsymbol{T}$ is modelled through the vector $\delta\boldsymbol{\pi}$. This approach allows us to exploit the advantages of a matrix representation for the rover poses, but maintain a probability distribution over a vector space. Similarly, we can solve for perturbations in a local vector space, and use (4.17) as a constraint-sensitive update to the transformation matrices during the Gauss-Newton iterations.

For homogeneous vectors, we operate directly on the position vector by considering additive perturbations of the form

$$\boldsymbol{p} = \overline{\boldsymbol{p}} + \mathbf{D}\,\delta\boldsymbol{\epsilon}, \tag{4.18}$$

where $\delta\boldsymbol{\epsilon}$ is a small 3×1 noise vector, $\overline{\boldsymbol{p}}$ is a noise-free homogeneous feature vector, and $\mathbf{D}$ is a dilation matrix defined as

$$\mathbf{D} := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}. \tag{4.19}$$

Since we restrict our perturbation to the 3×1 position vector, we do not need to enforce any additional constraints.

In summary, we utilize a singularity-free matrix representation for our state, but perform perturbations in a local vector space. This is incorporated into the batch state estimation algorithms by defining the state perturbation vector, $\delta\boldsymbol{\theta}$, to be composed of the two perturbation types, $\delta\boldsymbol{\pi}$ and $\delta\boldsymbol{\epsilon}$, and replacing the update step, $\overline{\boldsymbol{\theta}} \mapsfrom \overline{\boldsymbol{\theta}} + \delta\boldsymbol{\theta}^{\star}$, with (4.17) and (4.18) during estimation. Since the Gauss-Newton algorithm only requires a vector space for the perturbations, these definitions appropriately extend the estimation algorithms to the 3D domain.

4.4 Measurement Model Linearization

Finally, we utilize the perturbations to obtain linearized measurement error models for each Gauss-Newton iteration. These perturbations are summarized by

$$\boldsymbol{T}_{k,k_0} = e^{-\delta\boldsymbol{\pi}^{\boxplus}_{k,k_0}} \overline{\boldsymbol{T}}_{k,k_0} \approx \left(\mathbf{1} - \delta\boldsymbol{\pi}^{\boxplus}_{k,k_0}\right) \overline{\boldsymbol{T}}_{k,k_0}, \tag{4.20a}$$

$$\boldsymbol{p}^{j,k_0}_{k_0} = \overline{\boldsymbol{p}}^{j,k_0}_{k_0} + \mathbf{D}\,\delta\boldsymbol{\epsilon}^{j,k_0}_{k_0}, \tag{4.20b}$$

where we have kept only the first-order term in the Taylor-series expansion of the matrix exponential map. It should be noted that this approximation is only used to compute the linearizations, while the updates are performed using the exact exponential map.

We consider two common measurement models in this section, which are employed in the following chapters. The first is a generic camera model where a point in space is projected into the local sensor frame, and the second is a pose-to-pose relation for odometry measurements.

4.4.1 Camera Model

Using transformation matrices and homogeneous coordinates, the generic camera model is expressed as

$$\mathbf{z}_{jk} := \mathbf{f}\left(\boldsymbol{T}_{k,k_0} \boldsymbol{p}^{j,k_0}_{k_0}\right), \tag{4.21}$$

where $\boldsymbol{T}_{k,k_0}$ and $\boldsymbol{p}^{k,k_0}_{k_0}$ are state variables, $\mathbf{f}(\cdot)$ is the sensor model, and $\mathbf{z}_{jk}$ is the measurement. This model is composed of two nonlinearities. The first is the transformation of a point into the local frame, and the second is the sensor model. For 3D estimation, the challenging component is the point transformation, since it involves 3D rotations. We conduct this linearization by substituting in the perturbations defined above to get

$$\begin{aligned}\boldsymbol{p}_k^{j,k} &= \boldsymbol{T}_{k,k_0}\boldsymbol{p}_{k_0}^{j,k_0} \\ &\approx \left(\mathbf{1} - \delta\boldsymbol{\pi}^{\boxplus}\right)\overline{\boldsymbol{T}}_{k,k_0}\left(\boldsymbol{p}_{k_0}^{j,k_0} + \mathbf{D}\,\delta\boldsymbol{\epsilon}_{k_0}^{j,k_0}\right),\end{aligned} \tag{4.22}$$

and expand the expression while neglecting products of small terms to produce

$$\boldsymbol{p}_k^{j,k} \approx \underbrace{\overline{\boldsymbol{T}}_{k,k_0}\overline{\boldsymbol{p}}_{k_0}^{j,k_0}}_{\overline{\boldsymbol{p}}_k^{j,k}} - \delta\boldsymbol{\pi}_{k,k_0}^{\boxplus}\overline{\boldsymbol{T}}_{k,k_0}\overline{\boldsymbol{p}}_{k_0}^{j,k_0} + \overline{\boldsymbol{T}}_{k,k_0}\mathbf{D}\,\delta\boldsymbol{\epsilon}_{k_0}^{j,k_0}. \tag{4.23}$$

Finally, we employ (4.13) and group similar terms to obtain

$$\boldsymbol{p}_k^{j,k} \approx \overline{\boldsymbol{p}}_k^{j,k} + \begin{bmatrix}\overline{\boldsymbol{p}}_k^{j,k\boxminus} & \overline{\boldsymbol{T}}_{k,k_0}\mathbf{D}\end{bmatrix}\begin{bmatrix}\delta\boldsymbol{\pi}_{k,k_0} \\ \delta\boldsymbol{\epsilon}_{k_0}^{j,k_0}\end{bmatrix}. \tag{4.24}$$

Using the chain rule, this result can be multiplied with the Jacobian of the sensor model, $\mathbf{F}_{jk}$, to produce the linearized camera error expression

$$\mathbf{e}_{jk} \approx \mathbf{z}_{jk} - \mathbf{f}(\overline{\boldsymbol{p}}_k^{j,k}) - \mathbf{F}_{jk}\begin{bmatrix}\overline{\boldsymbol{p}}_k^{j,k\boxminus} & \overline{\boldsymbol{T}}_{k,k_0}\mathbf{D}\end{bmatrix}\begin{bmatrix}\delta\boldsymbol{\pi}_{k,k_0} \\ \delta\boldsymbol{\epsilon}_{k_0}^{j,k_0}\end{bmatrix}, \quad \mathbf{F}_{jk} := \left.\frac{\partial\mathbf{f}}{\partial\boldsymbol{p}}\right|_{\overline{\boldsymbol{p}}_k^{j,k}}. \tag{4.25}$$

4.4.2 Odometry

An odometry measurement is composed of a relative transformation between poses, which we define as

$$\boldsymbol{T}_{k+1,k} := \boldsymbol{T}_{k+1,k_0}\boldsymbol{T}_{k,k_0}^{-1}, \tag{4.26}$$

where $\boldsymbol{T}_{k+1,k_0}$ and $\boldsymbol{T}_{k,k_0}$ are state variables, and $\boldsymbol{T}_{k+1,k}$ is the measurement. This is more difficult to linearize due to the presence of multiple transformation matrices. We begin by noting the fact that we are interested in linearizing an error term, rather than linearizing the more general pose-to-pose transformation. This observation allows us to define

$$\boldsymbol{T}_{k+1,k}\boldsymbol{T}_{k,k_0}\boldsymbol{T}_{k+1,k_0}^{-1} =: e^{-\mathbf{e}_{k+1,k}^{\boxplus}} \approx \mathbf{1} - \mathbf{e}_{k+1,k}^{\boxplus}, \tag{4.27}$$

where we make the assumption that the error is small. For use in the estimation algorithms, we desire a linearized expression for $\mathbf{e}_{k+1,k}$, which captures the transformation estimate error in vector form. We accomplish this by substituting in the perturbations and neglecting products of small terms to obtain

$$\begin{aligned}\mathbf{1} - \mathbf{e}_{k+1,k}^{\boxplus} &\approx \boldsymbol{T}_{k+1,k}\left(\mathbf{1} - \delta\boldsymbol{\pi}_{k,k_0}^{\boxplus}\right)\overline{\boldsymbol{T}}_{k,k_0}\overline{\boldsymbol{T}}_{k+1,k_0}^{-1}\left(\mathbf{1} + \delta\boldsymbol{\pi}_{k+1,k_0}^{\boxplus}\right) \\ &\approx \boldsymbol{T}_{k+1,k}\overline{\boldsymbol{T}}_{k,k_0}\overline{\boldsymbol{T}}_{k+1,k_0}^{-1} - \boldsymbol{T}_{k+1,k}\delta\boldsymbol{\pi}_{k,k_0}^{\boxplus}\overline{\boldsymbol{T}}_{k,k_0}\overline{\boldsymbol{T}}_{k+1,k_0}^{-1} \\ &\quad + \boldsymbol{T}_{k+1,k}\overline{\boldsymbol{T}}_{k,k_0}\overline{\boldsymbol{T}}_{k+1,k_0}^{-1}\delta\boldsymbol{\pi}_{k+1,k_0}^{\boxplus},\end{aligned} \tag{4.28}$$

and strategically inserting a series of transformations into the second term to produce

$$\begin{aligned}\mathbf{1}-\mathbf{e}_{k+1,k}^{\boxplus} \approx\ & \boldsymbol{T}_{k+1,k}\overline{\boldsymbol{T}}_{k,k_0}\overline{\boldsymbol{T}}_{k+1,k_0}^{-1} - \boldsymbol{T}_{k+1,k}\overline{\boldsymbol{T}}_{k,k_0}\overline{\boldsymbol{T}}_{k+1,k_0}^{-1}\overline{\boldsymbol{T}}_{k+1,k_0}\overline{\boldsymbol{T}}_{k,k_0}^{-1}\delta\boldsymbol{\pi}_{k,k_0}^{\boxplus}\overline{\boldsymbol{T}}_{k,k_0}\overline{\boldsymbol{T}}_{k+1,k_0}^{-1} \\ & + \boldsymbol{T}_{k+1,k}\overline{\boldsymbol{T}}_{k,k_0}\overline{\boldsymbol{T}}_{k+1,k_0}^{-1}\delta\boldsymbol{\pi}_{k+1,k_0}^{\boxplus}. \end{aligned} \tag{4.29}$$

This allows us to utilize (4.14), and group similar terms to get

$$\mathbf{1}-\mathbf{e}_{k+1,k}^{\boxplus} \approx \boldsymbol{T}_{k+1,k}\overline{\boldsymbol{T}}_{k,k_0}\overline{\boldsymbol{T}}_{k+1,k_0}^{-1}\left(\mathbf{1}-\left(\left(\overline{\boldsymbol{T}}_{k+1,k_0}\overline{\boldsymbol{T}}_{k,k_0}^{-1}\right)^{\boxdot}\delta\boldsymbol{\pi}_{k,k_0}\right)^{\boxplus}+\delta\boldsymbol{\pi}_{k+1,k_0}^{\boxplus}\right). \tag{4.30}$$

Finally, we make the assumption that the difference between the transformation measurement and our current estimate is small (and known), such that

$$\boldsymbol{T}_{k+1,k}\overline{\boldsymbol{T}}_{k,k_0}\overline{\boldsymbol{T}}_{k+1,k_0}^{-1} =: e^{-\bar{\mathbf{e}}_{k+1,k}^{\boxplus}} \approx \mathbf{1}-\bar{\mathbf{e}}_{k+1,k}^{\boxplus}. \tag{4.31}$$

Expanding and neglecting products of small terms produces

$$\mathbf{1}-\mathbf{e}_{k+1,k}^{\boxplus} \approx \mathbf{1}-\bar{\mathbf{e}}_{k+1,k}^{\boxplus}-\left(\left(\overline{\boldsymbol{T}}_{k+1,k_0}\overline{\boldsymbol{T}}_{k,k_0}^{-1}\right)^{\boxdot}\delta\boldsymbol{\pi}_{k,k_0}\right)^{\boxplus}+\delta\boldsymbol{\pi}_{k+1,k_0}^{\boxplus}, \tag{4.32}$$

and using the property that $(\cdot)^{\boxplus}$ is linear results in the linearized transformation error expression

$$\mathbf{e}_{k+1,k} \approx \bar{\mathbf{e}}_{k+1,k} - \begin{bmatrix}\mathbf{1} & -\left(\overline{\boldsymbol{T}}_{k+1,k_0}\overline{\boldsymbol{T}}_{k,k_0}^{-1}\right)^{\boxdot}\end{bmatrix}\begin{bmatrix}\delta\boldsymbol{\pi}_{k+1,k_0} \\ \delta\boldsymbol{\pi}_{k,k_0}\end{bmatrix}. \tag{4.33}$$

This completes our linearized measurement error model derivations.

4.5 Summary

In summary, we have addressed how to extend the batch state estimation algorithms from Chapters 2 and 3 to the 3D domain by making modifications to three key areas. These modifications are employed in the following chapters, and are summarized as follows:

1. Define the state to be a collection of rover poses represented by transformation matrices, $\boldsymbol{T}_{k,k_0}$, and feature positions represented by homogeneous vectors, $\boldsymbol{p}_{k_0}^{j,k_0}$.
2. Employ constraint-sensitive perturbations and updates at each Gauss-Newton iteration, using (4.20).
3. Depending on the measurement types available, utilize (4.25) and (4.33) as linearized measurement error models, and redefine $\mathbf{z}_i$, $\mathbf{h}_i(\overline{\boldsymbol{\theta}})$, $\mathbf{H}_i$, and $\delta\boldsymbol{\theta}$ appropriately.

Chapter 5

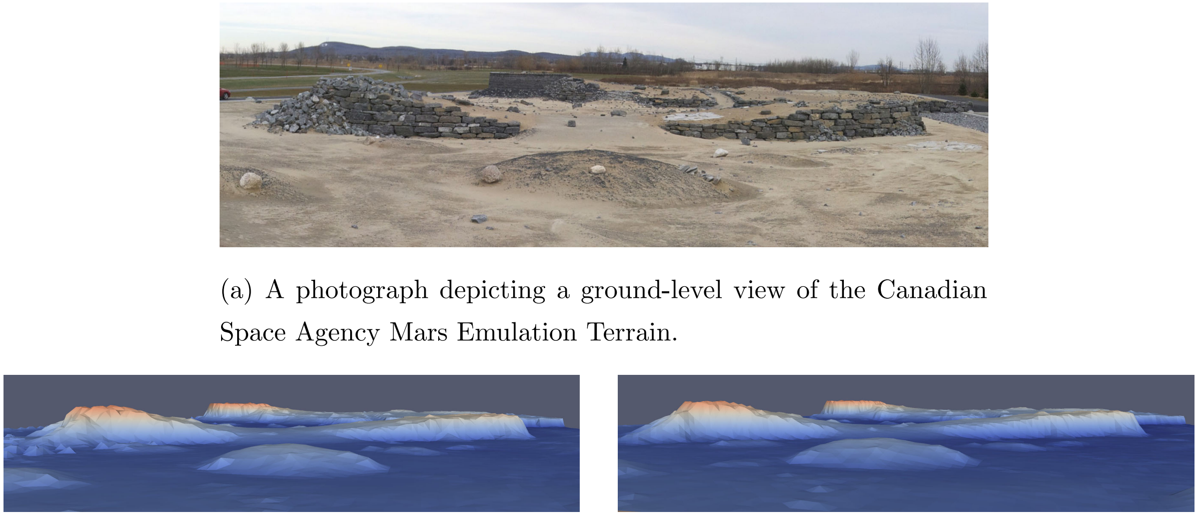
Planetary Worksite Mapping

5.1 Introduction

In this chapter, we present a robust 3D SLAM framework suitable for generating maps of a planetary worksite environment. This framework combines the mathematical preliminaries presented in Chapters 2 and 4 with other algorithmic components to achieve real-world operation.

As stated in the introduction, rover-generated planetary worksite maps are required for operations such as site selection, site preparation, and base construction. This was shown to be feasible by Fong et al. (2008a), who used a rover-mounted laser to perform systematic site surveys at the Haughton Crater. The resulting maps improved remote operator performance by enhancing situational awareness, and enabled high-duty exploration cycles. Additional mission concepts enabled by mapping technology were also identified, such as resource prospecting and reconnaissance (Fong et al., 2008b). However, GPS was used for localization in these experiments. While GPS produced accurate maps, planetary infrastructure that is able to provide similar levels of accuracy currently does not exist. Therefore, it is the intent of this chapter to present an infrastructure-free framework for autonomously producing planetary worksite maps.

We preview the results of our mapping framework in Figure 5.1, where we were able to produce an estimated map of an unstructured environment that closely resembles the true terrain. The algorithmic components required to produce these maps are described in detail in this chapter. In this work, we focus on the offline mapping scenario, which considers the problem of generating a detailed map after an exploratory traverse is conducted. Though augmentations would be required for online operation, accurate de-



(a) A photograph depicting a ground-level view of the Canadian Space Agency Mars Emulation Terrain.

(b) Reference map (using DGPS for localization).

(c) Estimated map (our mapping framework).

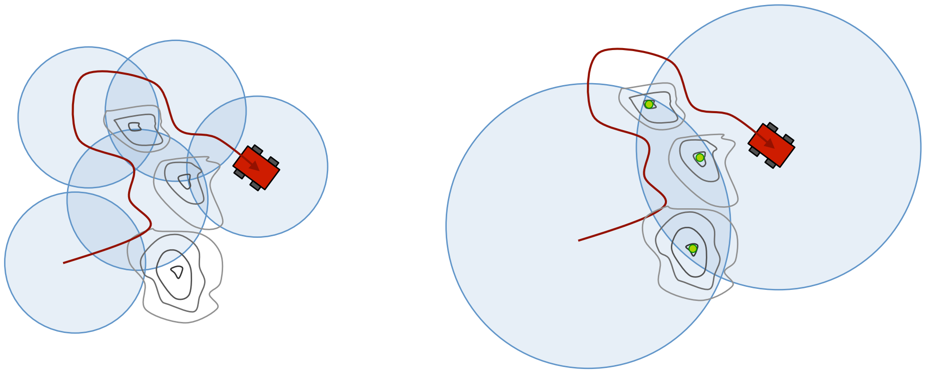
Figure 5.1: A sample set of images that allows for qualitative evaluation of the performance of the proposed mapping framework. All three images are rendered from approximately the same viewpoint, with a photograph in (a), and meshes created from the reference and estimated point clouds in (b) and (c), respectively. The mesh colour indicates elevation, with blue corresponding to low, and red to high. It can be seen that the estimated map closely resembles the photo and the reference render of the terrain.

termination of the rover poses is required to align scans and construct a map. Therefore, we consider the joint task of localization and mapping of rough terrain in an unknown environment as a 3D SLAM problem. Online localization suitable for exploration is addressed in Chapter 6.

As discussed previously, poor ambient lighting conditions necessitate the use of actively illuminated sensors such as laser rangefinders. One possible approach for mapping using laser data is to adapt the appearance-based SLAM techniques used with passive cameras to laser scanners by utilizing intensity images (McManus et al., 2011). While we employ this approach for VO in the following chapter, it is currently unclear if conventional appearance-based feature detectors and descriptors from the computer vision literature are robust to large viewpoint changes. Since numerous loop closures will occur during worksite exploration, we opt to use the range data instead. There are two main approaches in the current literature for aligning point cloud data. The first operates on the dense point clouds (Rusinkiewicz and Levoy, 2001), and the second utilizes a sparse geometric feature representation (Heider et al., 2011).

In the case of dense matching, point clouds can be aligned in a pairwise fashion using algorithms such as Iterative Closest Point (ICP) (Besl and McKay, 1992), applied sequentially to scans as they are collected throughout the rover's traverse (Nüchter et al., 2007). Though simple in description, dense alignment algorithms have demonstrated impressive results, producing very accurate estimates under normal operating conditions. However, since an initial estimate is required to initialize the algorithm, these methods are highly sensitive to the quality of the initial guess. In practice, these initial guesses are typically provided through inertial sensor measurements or wheel odometry, and ICP used for refinement. These measurements tend to be very uncertain over long distances, which is unaccounted for by dense alignment algorithms. As a result, the poor initial guesses start the algorithm in the wrong location, and result in convergence to an incorrect local minimum. Though these failures can be reduced by limiting the spacing between scans, this implies a certain type of rover traverse. Fundamentally, dense matching methods are more amenable to the incremental mapping scenario, where short-range scans are obtained frequently. Due to power limitations, it is more likely that a planetary rover will conduct a surveying-style traverse, where long-range scans are obtained infrequently. An illustration depicting the two different exploration scenarios is provided in Figure 5.2, where we see that ICP would be inappropriate for the surveying approach. Furthermore, since the matches are computed pairwise, alignment errors can compound, which can lead to a globally inconsistent map. These inconsistencies can be hard to detect, since an additional layer is required to recognize loop closures.

On the other hand, a sparse-feature-based approach can be taken, where geometrically distinct interest points are extracted from each scan. These distinctive points can include peaks in the terrain (Carle et al., 2010), or regions of high curvature (Bakambu et al., 2006b). This approach obtains a reduced representation of the scan, resulting in computational savings. Furthermore, matching these points provides implicit loop closure. Unfortunately, geometric feature matching is not a simple task. For point-to-point matching, a descriptor for the local structure must be constructed for each point, so that it can be recognized in multiple scans. A wide variety of geometric descriptors have been proposed in the literature, including spin images (Johnson, 1997), point fingerprints (Sun et al., 2003), and local shape descriptors (Taati et al., 2007). However, in the case of distinctive geometric features such as large rocks or tall peaks, occlusions result in insufficient data to construct an appropriate descriptor. Finding suitable geometric features and descriptors for natural terrain remains an active research area.



(a) Incremental mapping.

(b) Surveying.

Figure 5.2: An illustration depicting different traverse characteristics for a rover navigating around a small ridge. In (a), an incremental mapping approach is taken where numerous short-range scans are taken all around the ridge. This scanning approach is more amenable to dense matching methods, because sufficient overlap is maintained at all times. On the other hand, (b) depicts a surveying-style scan procedure, where long-range scans are obtained infrequently. In this case, the larger-scale peak features (indicated by the green circles) are more appropriate for accurate alignment.

In our framework, we utilize a hybrid approach to feature matching that gains the advantages of both the sparse-feature and dense-data representations. The use of sparse features reduces the computational requirements, and descriptors are avoided by utilizing the spatial relation between groups of features (i.e., feature constellations). This approach is common in attitude determination algorithms that utilize a star tracker (Liebe, 1995). However, incorrect matches can arise from using the spatial relations alone, so the dense data are reintroduced for the verification of match hypotheses. In this hybrid approach, the value of the rich dense data is retained, but utilized sparingly.

In case of feature scarcity, we also consider pose-to-pose measurements in the form of odometry. These odometry measurements could come from the laser-based VO algorithm presented in the following chapter, or other sources of sequential pose estimates such as wheel encoders. Though these measurements may be too noisy to be used as initial guesses for ICP, an appropriate consideration of the measurement uncertainty only serves to improve the alignment estimate in our framework. Finally, operation in a confined worksite environment will involve a large number of loop closures. As a result, we consider the SLAM problem as a batch alignment problem, which maintains the global consistency of the estimated map.

While the measurement models used in alignment are formulated to account for noise corruption, invariably, erroneous sensor measurements can still occur. Outlier measurements that do not fit the sensor model can arise due to incorrect data associations, spurious features, or slippage in odometry. In our proposed system, we improve the robustness by considering the possibility of outliers in both measurement types, and reject them within the batch alignment framework (Tong and Barfoot, 2011a, 2013a).

In summary, we present a complete 3D SLAM framework in this chapter, which is able to robustly create globally consistent maps of natural, unstructured 3D terrain. This framework is the first laser-based 3D SLAM algorithm to utilize a combination of sparse features, dense data, and odometry measurements in a batch alignment framework, and incorporates heterogeneous outlier rejection for robustness. Furthermore, this framework was validated using data gathered at two planetary analogue test facilities. Though the emphasis of this work was an offline proof-of-concept implementation, some timing information is provided to help identify the bottlenecks for improvement.

Distinguishing aspects of the intended scenario include the low scan overlap arising from the surveying-style traverse, and feature scarcity due to operation in natural terrain. This framework uniquely addresses both concerns by integrating various algorithmic components. The contents of this chapter have appeared in both Tong et al. (2011) and Tong et al. (2012a), and the framework has been utilized for human-guided exploration in Merali et al. (2012).

The remainder of the chapter is organized as follows. Section 5.2 details related work, and Sections 5.3 to 5.6 describe the proposed framework. This is followed by the experimental section, which contains a discussion of the evaluation metrics in Section 5.7, and the results at the two test facilities in Sections 5.8 and 5.9. Finally, a discussion is provided in Section 5.10, and concluding remarks are made in Section 5.11.

5.2 Related Work

In this section, we review related work on laser-based 3D SLAM in natural environments. Since our objective is worksite mapping, this review focuses on globally consistent methods, and adaptations of pairwise methods that can produce globally consistent maps.

Pairwise alignment of sequential scans of planetary analogue terrain has been conducted for 3D localization using variants of ICP by several research groups. Notable applications to rover navigation include the experiments conducted by Se et al. (2004)

in a quarry, the work by Bakambu et al. (2006a) for localization and planning, and the inclusion of surface-normal information by Gemme et al. (2012) to result in improved convergence. Bakambu et al. (2006b) also considered a feature-based approach, where regions of high curvature were described using both spin images (Johnson, 1997) and point fingerprints (Sun et al., 2003). However, these experiments were only conducted for a small range of motions, resulting in limited viewpoint change. While Wettergreen et al. (2009) successfully demonstrated dark navigation using dense alignment and a combination of an inertial measurement unit (IMU) and a downwards-facing optical sensor for initial pose estimation, the motion between scans was limited to a maximum of 3m in translation, and 10° in rotation.

Alternative scan representations used for pairwise alignment include the Normalized Distributions Transform (NDT) (Magnusson et al., 2007), which provides a reduced scan representation by discretizing and describing the scan volume. While this representation was later extended to provide loop closure detection (Magnusson et al., 2009), these approaches were designed for localization in underground mines, where walls are prevalent. Similarly, the use of planar surface patches (Pathak et al., 2010) for 3D mapping of urban disaster sites utilizes inherent geometry. In both cases, the existence of man-made structures supported strong priors that were exploited for scan registration (Johnson-Roberson et al., 2010). In unstructured, natural terrain, planar elements are rare.

Since dense alignment methods are sensitive to the quality of the initial guess, a possible approach would be to use a feature-based method for initial alignment, and dense data for refinement. This approach was utilized by King et al. (2005), where a combination of interest points extracted from intensity images was used to generate transformation hypotheses, and an ICP variant was used for alignment of urban imagery. The framework presented in this chapter could serve the purpose of providing initial guesses for dense refinement. Other hierarchical alignment approaches used for urban environments include coarse-to-fine scan registration (Brenner et al., 2008), and skyline-based methods (Nüchter et al., 2011). However, a previous comparison conducted by Carle et al. (2010) between a skyline-based method (Cozman et al., 2000) and 3D peak features for long-range localization found that closer-range 3D information useful for alignment was better captured through the use of interest points.

Due to alignment errors, compounding the estimates produced by pairwise methods can lead to a global inconsistency. An approach aimed at adapting these pairwise methods to create globally consistent maps introduced the concept of incrementally building a metascan (Chen and Medioni, 1992). Unfortunately, this method does not spread out the error, and is order dependent. Alternative multiview registration algorithms aligned all of the dense data directly (Cunnington and Stoddart, 1999; Williams and Bennamoun, 2001), but the accuracy of point correspondences were still dependent on the quality of the initial guesses. Furthermore, operating directly on all of the dense data can be computationally intractable for large data sets (Pulli, 1999). While segmenting the traverse into submaps (Fairfield et al., 2010) can limit the growth of local error, global consistency was still not addressed.

Alternatively, the 6DSLAM work (Nüchter et al., 2007) first utilized ICP to compute the pairwise alignments, and used a global relaxation technique (Borrmann et al., 2008) for refinement. Though this approach distributes the pairwise errors over the entire map, the overall performance was still limited by the quality of the initial guesses. This is attributed to the fact that pairwise ICP still remains at the core of the approach. Furthermore, as loop closures were indicated by a simple distance criterion, long loops were not detected (Wulf et al., 2008). In our framework, we utilize a combination of sparse features and odometry measurements for batch alignment. The use of sparse features provides implicit loop closure detection, and the inclusion of odometry adds flexibility in the distance travelled between scans.

Finally, the Multi-frame Odometry-compensated Global Alignment (MOGA) (Carle et al., 2010) algorithm for long-range localization matched ground-based laser scans to orbital maps using a combination of sparse features and odometry measurements in a batch alignment formulation. Feature measurement outlier rejection was applied between the scans and the orbital map in a pairwise fashion, using the Random Sample and Consensus (RANSAC) (Fischler and Bolles, 1981) algorithm to detect incorrect data associations. This was proven through field tests in the Canadian High Arctic, achieving position errors of less than 100m on a 10km by 10km orbital map with a 30m resolution.

We build upon the successful results of Carle et al. (2010) by adapting the algorithms to a relatively short-range scenario. The dependence on a prior orbital map is removed from the alignment algorithm, and the hybrid data association is reformulated for local map matching. Furthermore, the robustness of the framework is improved by considering the possibility of outliers in both measurement types through the incorporation of heterogeneous measurement outlier rejection (Tong and Barfoot, 2011a, 2013a). Finally, since the objective of the work is to create a worksite map, we also introduce an automatic verification step after alignment to ensure the quality of the final map.

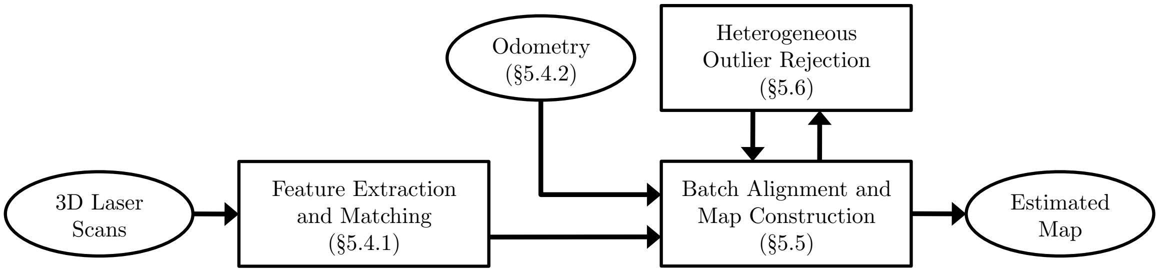


Figure 5.3: The laser-based 3D SLAM architecture utilized in this chapter, with the relevant sections indicated. Feature measurements extracted from the 3D laser scans are used alongside odometry measurements in a batch alignment algorithm, which incorporates heterogeneous outlier rejection for robustness. After convergence, the pose estimates are validated, and the scans are overlaid to construct the final map.

5.3 Architecture Overview

In consideration of the slow nature of conventional long-range 3D laser scanners such as surveying lidars or planar lasers mounted on panning units, a *stop-scan-go* approach is taken, where the rover traverses over a distance, and occasionally stops to conduct a scan of the terrain. Though high-rate laser sensors may be available, these sensors do not offer the same level of accuracy and range. Furthermore, this approach ensures local consistency of the point cloud by avoiding the complexity of motion distortion within a single scan. We elaborate upon these issues in Chapter 6.

Our mapping architecture for this scenario is illustrated in Figure 5.3. Given a series of point clouds, distinct interest points are extracted and matched between scans. These feature observations are then considered alongside odometry measurements in a batch alignment algorithm, and heterogeneous outlier rejection is performed for robustness. Upon convergence, the pose estimates are validated through an automatic verification test, and the scans are overlaid to construct the final map.

The sections to follow discuss each algorithmic component in further detail. Section 5.4 explains how the feature measurements are extracted from the scans, as well as the advantages of incorporating odometry into the alignment. Section 5.5 details the batch alignment formulation, including the measurement models, estimate reference frame selection, and the automatic verification step for map construction. Finally, Section 5.6 describes how heterogeneous outlier classification is incorporated into the alignment algorithm, providing improved robustness towards outlier measurements.

5.4 Measurements

In this section, we detail the measurement generation process that provides the inputs for scan alignment. These measurements include feature positions extracted from laser scans, and odometry estimates obtained during navigation.

5.4.1 Natural Feature Extraction and Matching

Due to the limitations of the dense data and sparse feature approaches highlighted in Section 5.1, we utilize a hybrid approach that gains the advantages of both representations. This approach adapts the methods of Carle et al. (2010) to the local mapping scenario, where a single sensor is used to obtain point cloud data. As a result, the feature matching procedure is simplified.

In our implementation, peaks in the terrain are chosen to serve as the distinctive interest points due to their ease of extraction and visibility at long ranges. As the choice of peaks as features may limit the feature extraction and matching approach presented in this section to 2.5D environments, an alternative approach set of features may need to be used in fully 3D scenarios. The following sections describe each step in the peak extraction and matching process.

5.4.1.1 Feature Detection

Given a scan of the terrain, the dense point clouds are first rotated so that the $+z$ direction corresponds to the opposite direction of the gravity vector, as depicted by the coordinate frame in Figure 5.4. These pitch and roll corrections can be provided by an inclinometer, which is a low-cost sensor that is easily incorporated into a sensor payload. Though this could fail in situations where a wheel is propped up by a rock, we heuristically assume that scans are obtained from safe configurations.

If an inclinometer is unavailable, an estimate can be obtained by applying heuristics based on prior knowledge of the terrain. For example, a possible heuristic could be that the majority of the terrain is flat. Therefore, a plane-fit conducted over all of the data can determine the overall scan normal, which should correspond to the terrain normal. While we found that this heuristic worked in the majority of cases, it failed with scans taken on large slopes, where the slope occluded the majority of the scan. As a result, we opted for the simpler solution of integrating an onboard inclinometer.

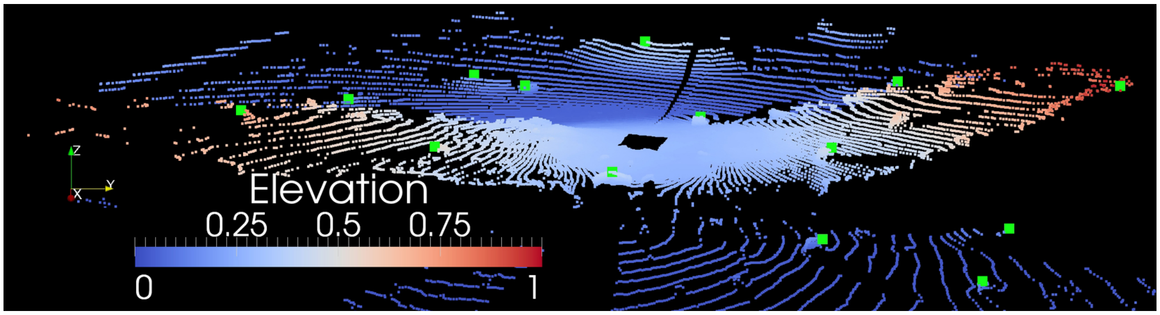


Figure 5.4: A sample point cloud obtained by a 3D laser rangefinder in planetary analogue terrain, with the world coordinate frame depicted on the left side. The scan was taken from the centre of the image, with the rectangular cutout corresponding to the rover's footprint, and the green squares indicating the detected peak features. This point cloud was levelled using the plane-fit technique, which succeeded because the majority of the data corresponded to the flat terrain. However, if occlusions dominate the scan, this technique could fail to identify the terrain normal correctly.

With each levelled scan, peaks are detected by simply considering the z-components. For each point, we define a local neighbourhood proportional to the scan resolution at that range. If the point has the largest z-value in its local neighbourhood, it is classified as a candidate peak. As can be seen in Figure 5.4, the two distinctive peaks characterized by the large elevation values are detected, as well as other smaller rocks in the scene. We avoid spurious features by performing additional tests to ensure that the peaks are sufficiently large, and that there are enough points in the local neighbourhoods for support. Due to these additional tests, it was found that nonmax suppression was unnecessary for ridge rejection. The computational requirements of this process can be reduced through scan decimation, or by filtering out points below a minimum height threshold.

Though the sparse feature approach represents the peaks as single points in space, in reality, there is uncertainty due to the landmark shape and sensor characteristics. In particular, the constant angular resolution of a laser rangefinder results in large point spacing when converted into Cartesian coordinates. Since this dominates the other sources of landmark position error, we conservatively assign the uncertainty of each interest point to be isotropic, and simply proportional to the scan resolution at the landmark range. This positional uncertainty is captured by the covariance matrix associated to each feature measurement.

5.4.1.2 Feature Matching

Matching peaks is a multistep process. For each pair of scans, constellations of features are matched to generate hypotheses. Next, the hypotheses are validated using the dense data, and finally, the valid hypotheses are resolved into individual feature associations.

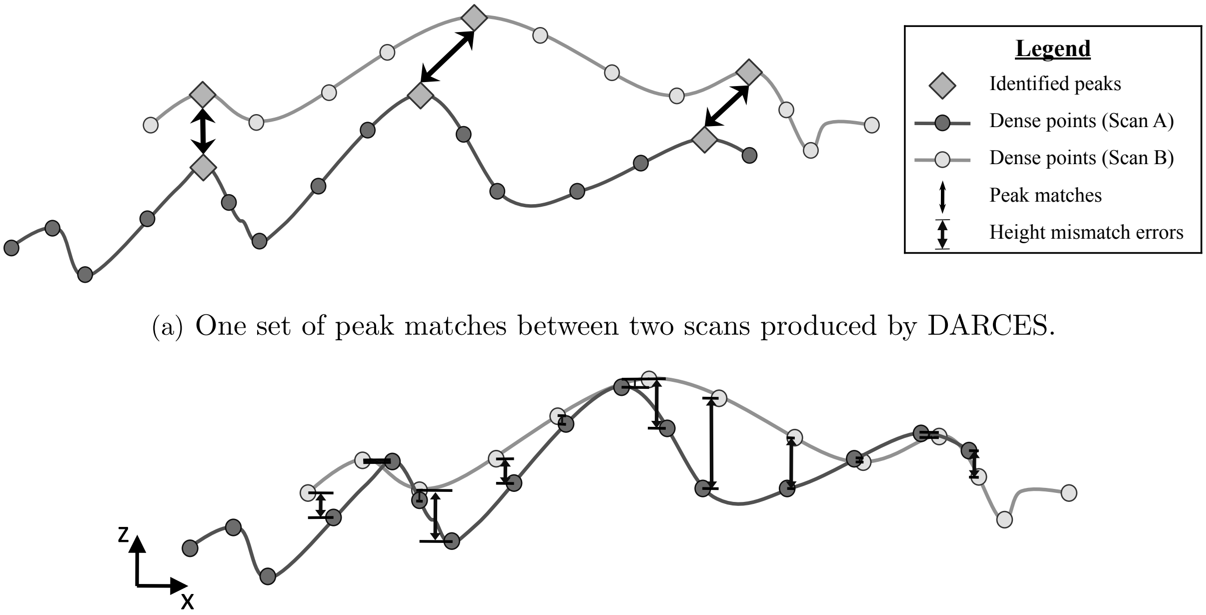
Constellation matching can be considered as seeking sets of similar triangles between scans. The Data-Aligned Rigidity-Constrained Exhaustive Search (DARCES) (Chen et al., 1999) algorithm provides a method to do this, while incorporating the position uncertainty of each measurement. However, due to the large uncertainty values, this tends to produce a large number of hypotheses. To reduce this number to only the correct hypotheses, a quality metric is computed for each pair of matching triangles.

The dense data are reintroduced at this stage for hypothesis validation. For each matching triangle, an alignment hypothesis can be performed between the scans. Using the rough alignment, the dense data are compared, and a score is assigned based on the average mismatch in the z-component of the overlapping regions. Due to uneven point spacing, the points from each scan are not vertically aligned. Therefore, the mismatch is computed between a point and its nearest neighbour in (x, y) space. An illustration of the steps conducted to compute the dense data verification metric is provided in Figure 5.5. This metric assesses the quality of the rough alignment, while considering the non-uniform spacing of the two scans. A threshold is then applied on the scores, resulting in a set of valid hypotheses.

After obtaining a set of valid hypotheses, the feature associations are separated back into individual measurements. Due to measurement noise, some interest points may have multiple associations. As a result, the associations are resolved in order of match quality (i.e., based on the dense scan verification step).

The peak matching algorithm can be summarized as follows:

1. Match similar triangle feature constellations between two scans using DARCES.
2. For each matching triangle:
   (a) Compute an alignment hypothesis using the three matching points.
   (b) Perform the rough alignment, and overlay the dense scans.
   (c) Compute and store the dense verification metric for the overlapping region.
3. Threshold and keep only the matching triangles that meet a minimum score.
4. Resolve the matching triangles back into individual feature measurements.



(a) One set of peak matches between two scans produced by DARCES.

(b) The dense verification metric after the alignment hypothesis is performed.

Figure 5.5: Illustrations of the dense verification metric used to evaluate the quality of similar triangle matches between two scans. For clarity, a side profile viewpoint is provided, but in implementation, this process is conducted in 3D. In (a), we depict a set of three peak correspondences identified in two scan profiles. An alignment hypothesis is computed between these correspondences, producing the arrangement in (b). As can be seen, these peaks may not align exactly due to measurement noise, or other errors. A score is then computed based on the average height mismatch in the overlapping region, reflecting the quality of the hypothesis. Since the points are not vertically aligned, a nearest neighbour search is conducted to determine the corresponding height values.

5.4.2 Odometry

The incorporation of odometry measurements into the alignment problem provides flexibility in the rover traverse. Pose-to-pose scan relations are maintained not only in regions of feature scarcity, but also in situations without any scan overlap. For rover navigation, odometry measurements are typically used as feedback to guide the rover to the goal location. In the worksite mapping scenario, some of the rover's goals are the scan positions. Since these measurements are already available, it is straightforward to store them for later use. If these measurements are appropriately incorporated into the batch alignment algorithm, the presence of more information benefits alignment accuracy.

Though basic odometry can be provided through wheel encoders, a variety of methods are available in the literature for producing sequential 3D pose estimates. For example, while we are focusing on environments where varied lighting conditions can cause issues for passive cameras, stereo camera VO can still be useful over short distances. Alternatively, laser data can be constructed into intensity images (McManus et al., 2011), and used to compute VO estimates. This is addressed in the following chapter.

5.5 Batch Alignment

To maintain global consistency, we use a batch approach to solve the 3D SLAM problem. This approach makes use of the discrete-time batch Gauss-Newton algorithm presented in Chapter 2, and the 3D state parametrization described in Chapter 4.

As a reminder, the batch formulation casts the estimation problem as an optimization task, where we minimize an appropriate objective function using the Gauss-Newton algorithm. This minimization is performed by linearizing the measurement error models, solving for an optimal state perturbation, and iterating until convergence. While many details were presented in Chapters 2 and 4, we restate some key equations in this section for convenience.

5.5.1 System Models

We begin by defining the states we wish to estimate. In the SLAM problem, these states are the rover poses and the feature positions. As described in Chapter 4, we express these quantities of interest with respect to an estimate reference frame, $\underrightarrow{\mathcal{F}}_{k_0}$, parametrize the rover pose k with a transformation matrix, $\boldsymbol{T}_{k,k_0}$, and represent the position of feature j with a homogeneous coordinate vector, $\boldsymbol{p}_{k_0}^{j,k_0}$. For the Gauss-Newton algorithm, these states are perturbed using

$$\boldsymbol{T}_{k,k_0} = e^{-\delta\boldsymbol{\pi}_{k,k_0}^{\boxplus}}\overline{\boldsymbol{T}}_{k,k_0}, \tag{5.1a}$$

$$\boldsymbol{p}_{k_0}^{j,k_0} = \overline{\boldsymbol{p}}_{k_0}^{j,k_0} + \mathbf{D}\,\delta\boldsymbol{\epsilon}_{k_0}^{j,k_0}, \tag{5.1b}$$

where $\overline{(\cdot)}$ denotes the current state estimate, and $\delta\boldsymbol{\pi}_{k,k_0}$ and $\delta\boldsymbol{\epsilon}_{k_0}^{j,k_0}$ are the rover pose and feature position perturbations, respectively. These expressions are used to obtain linearized measurement error models, and to compute the optimal updates that progressively bring us closer to the minimal estimate.

Next, we define the measurement models using these state definitions[1]. Our feature measurements are extracted from the dense scans as positions with respect to the local rover frame. As a result, we define the feature measurements to be the position of feature j with respect to pose k:

$$\mathbf{z}_{jk} := \mathbf{f}\left(\boldsymbol{T}_{k,k_0}\boldsymbol{p}_{k_0}^{j,k_0}\right) + \mathbf{n}_{jk}, \quad \mathbf{n}_{jk} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{jk}), \tag{5.2}$$

where the noise parameter, $\mathbf{n}_{jk}$, is additive, zero-mean, and Gaussian-distributed, with a covariance of $\mathbf{R}_{jk}$. The camera model, $\mathbf{f}(\cdot)$, is a simple transformation from homogeneous coordinates to a standard (x, y, z) vector. That is,

$$\mathbf{f}(\boldsymbol{p}) := \frac{1}{p_4}\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}. \tag{5.3}$$

For measurement error linearization, we apply the linearized camera model expression from Chapter 4 to obtain

$$\mathbf{e}_{jk} \approx \mathbf{z}_{jk} - \mathbf{f}(\overline{\boldsymbol{p}}_k^{j,k}) - \mathbf{F}_{jk}\begin{bmatrix} \overline{\boldsymbol{p}}_k^{j,k^{\boxminus}} & \overline{\boldsymbol{T}}_{k,k_0}\mathbf{D} \end{bmatrix}\begin{bmatrix} \delta\boldsymbol{\pi}_{k,k_0} \\ \delta\boldsymbol{\epsilon}_{k_0}^{j,k_0} \end{bmatrix}, \quad \mathbf{F}_{jk} := \left.\frac{\partial\mathbf{f}}{\partial\boldsymbol{p}}\right|_{\overline{\boldsymbol{p}}_k^{j,k}}, \tag{5.4}$$

where the camera model Jacobian, $\mathbf{F}_{jk}$, is

$$\mathbf{F}_{jk} = \frac{1}{p_4}\begin{bmatrix} 1 & 0 & 0 & -\frac{p_1}{p_4} \\ 0 & 1 & 0 & -\frac{p_2}{p_4} \\ 0 & 0 & 1 & -\frac{p_3}{p_4} \end{bmatrix}. \tag{5.5}$$

The other measurement type, odometry, consists of a relative transformation between consecutive poses. This is expressed as

$$\boldsymbol{T}_{k+1,k} := e^{-\mathbf{w}_{k+1}^{\boxplus}}\boldsymbol{T}_{k+1,k_0}\boldsymbol{T}_{k,k_0}^{-1}, \quad \mathbf{w}_{k+1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k+1}), \tag{5.6}$$

where the noise parameter, $\mathbf{w}_{k+1}$, is zero-mean and Gaussian distributed with covariance $\mathbf{Q}_{k+1}$, and applied multiplicatively through the exponential map. Once again, the linearized error expression is obtained from Chapter 4, resulting in

$$\mathbf{e}_{k+1,k} \approx \overline{\mathbf{e}}_{k+1,k} - \begin{bmatrix} \mathbf{1} & -\left(\overline{\boldsymbol{T}}_{k+1,k_0}\overline{\boldsymbol{T}}_{k,k_0}^{-1}\right)^{\boxdot} \end{bmatrix}\begin{bmatrix} \delta\boldsymbol{\pi}_{k+1,k_0} \\ \delta\boldsymbol{\pi}_{k,k_0} \end{bmatrix}. \tag{5.7}$$

[1] The measurement models defined here are not the original equations presented in Tong et al. (2012a); however, in the interest of unifying the notation in this thesis, we present equivalent expressions in this section.

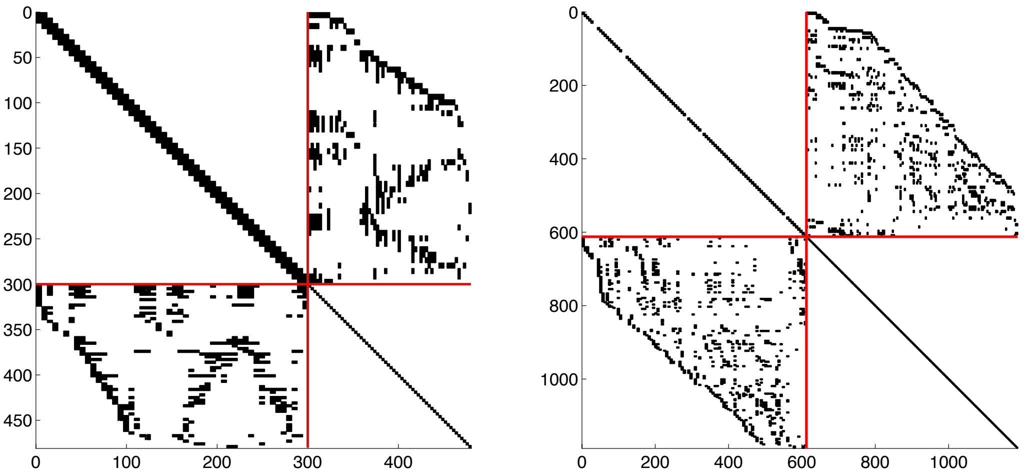
5.5.2 Optimization

To obtain the estimate that best matches the measurements, we simultaneously minimize all of the squared measurement errors, which are weighted by their inverse covariances to form a Mahalanobis distance (Mahalanobis, 1936). This results in the objective function

$$J := \frac{1}{2}\sum_{k=1}^{K-1} \mathbf{e}_{k+1,k}^T \mathbf{Q}_{k+1}^{-1} \mathbf{e}_{k+1,k} + \frac{1}{2}\sum_{k=1}^{K}\sum_{j=1}^{L} \mathbf{e}_{jk}^T \mathbf{R}_{jk}^{-1} \mathbf{e}_{jk}, \tag{5.8}$$

where K is the number of rover poses, and L is the number of features in the scene. Since this objective function is of the same form as (2.4), we can follow the Gauss-Newton procedure detailed in Chapter 2 to obtain the optimal solution.

Using (5.4) and (5.7), a linear system of equations is constructed during each Gauss-Newton iteration. As described in Chapter 2, this system is solved efficiently using the Schur complement. However, as depicted in Figure 5.6, the rover poses can outnumber the features in planetary terrain. In this case, we can condition out the rover poses instead, which results in a computational complexity of $O(L^3 + KL^2)$. After solving for the optimal perturbations, the state estimates are updated using the constraint-sensitive update equations (5.1), and the process repeats until convergence.



(a) UTIAS indoor rover test facility dataset.

(b) CSA Mars Emulation Terrain dataset.

Figure 5.6: The sparsity pattern of the batch alignment inverse covariance matrix, **A** (2.12), for both planetary analogue test facilities. The red lines indicate the partitioning between rover poses and feature positions. As can be seen, both the upper-left and lower-right blocks have an easily invertible form.

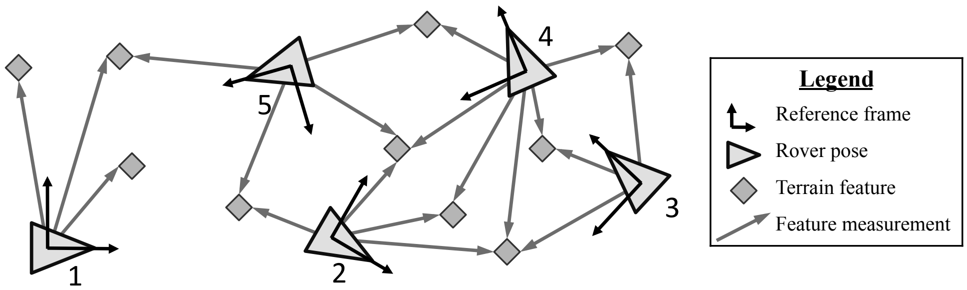


Figure 5.7: An illustration of a simple scenario where feature measurements are displayed along with reference frames attached to the rover poses. Due to an insufficient number of common measurements for 3D alignment, the first scan is disconnected from the rest. As a result, it would be a poor choice for the estimate reference frame. Instead, we heuristically choose Scan #4, since it is the most connected with the other poses.

5.5.3 Estimate Reference Frame Selection

In our formulation, the estimates produced by the batch alignment algorithm are expressed with respect to a single estimate reference frame, $\underrightarrow{\mathcal{F}}_{k_0}$. While this reference frame is conventionally chosen to be the first scan pose, the frame selection can have a large impact on the estimate quality. For example, the number of measurements may be insufficient to produce an estimate for every scan. Since every pose is expressed with respect to the reference frame, the links formed by measurements are essential to produce a pose estimate. If the reference frame is chosen incorrectly, an important part of the map may be unavailable. An illustration of such a situation is provided in Figure 5.7.

In the illustration, we depict a simple scenario with five possible estimate reference frames attached to the rover poses and their associated feature measurements. Since a minimum of three common features are required for 3D alignment, Scan #1 is disconnected from the rest. Therefore, if Scan #1 was naively chosen as the estimate reference frame, alignment estimates for the remaining four scans would be impossible.

The choice of reference frame should depend on the desired utility. For example, online rover navigation requires the current pose and the local terrain. Therefore, the latest pose should be chosen. However, for mapping, we seek to produce the largest globally consistent map. In this case, a well-connected pose is desirable, as it is likely to produce an estimate that is inclusive of the other scans. As a result, we heuristically choose the scan with the largest number of common features with the other scans to be our estimate reference frame. In our simple illustration, Scan #4 would be chosen.

5.5.4 Final Map Construction

Finally, we consider the end product of our framework, the map. Though a map can be constructed by simply overlaying the dense point clouds according to the rover pose estimates, we would like to emphasize that our goal is not localization of the rover; rather, it is to create an accurate map. As a map is typically used for further decision making, we believe that an accurate map with limited range is preferable to an inaccurate map with full coverage. Since the performance of an estimation algorithm largely depends on the quality of the measurements available, it is possible that an estimated pose is very uncertain, inaccurate, or simply unknown.

We address these concerns by introducing a post-alignment automatic verification check to ensure that the alignment estimate produced for each rover pose is suitable for inclusion into the final map. This is accomplished by applying a simple threshold on the pose uncertainty values, which are obtained from the block-diagonal entries of the estimate covariance matrix (2.11). In the conventional state ordering, the first K 6×6 diagonal blocks correspond to the rover poses, and the remaining L 3×3 diagonal blocks to the feature positions. Using these block-diagonal quantities, if we define the covariance associated to the kth pose estimate as $\mathbf{\Sigma}_k$, pose estimate k is valid if

$$\left(\sigma^2_{k,11} + \sigma^2_{k,22} + \sigma^2_{k,33}\right) < s^2_{\text{trans}}, \tag{5.9}$$

$$\left(\sigma^2_{k,44} + \sigma^2_{k,55} + \sigma^2_{k,66}\right) < s^2_{\text{rot}}, \tag{5.10}$$

where $\sigma^2_{k,ii}$ is the ith diagonal entry of $\mathbf{\Sigma}_k$, s^2_{trans} is a user-defined translational uncertainty threshold, and s^2_{rot} is a user-defined rotational uncertainty threshold. This check introduces additional robustness to the framework, compensating for inadequate performance at the feature extraction and mapping stage.

5.6 Heterogeneous Measurement Outlier Rejection

While the measurement models in the alignment algorithm are formulated to account for noise corruption, the presence of outlier measurements could result in catastrophic failures. Due to the repetitive terrain and the possibility of incorrect data associations, outliers are inevitable in a real-world scenario. In our framework, robustness towards outliers in both odometry and feature measurements is obtained by adapting the Normalized Innovation Squared (NIS) test (Bar-Shalom et al., 2002) from the data association literature to handle heterogeneous measurements (Tong and Barfoot, 2011a, 2013a).

Furthermore, iterative reclassification is used to improve the classification accuracy during alignment. This is similar to M-estimation (Zhang, 1997), but we account for the estimator uncertainty. As a result, more accurate classifications are obtained using this method than through conventional M-estimation or RANSAC alone. A brief derivation of the NIS test is presented in Appendix B, but the key results are provided here. For clarity, we present vector space expressions, but this test is easily adapted to the 3D domain using the methods described in Chapter 4.

The NIS test is a statistical criterion for classifying outlier measurements given a current hypothesis model. We assume that the hypothesis model is composed of estimated values for the state, $\overline{\boldsymbol{\theta}}$, and an associated covariance, $\overline{\boldsymbol{\Sigma}}$, such that the true state, $\boldsymbol{\theta}$, is modelled by the following relation:

$$\overline{\boldsymbol{\theta}} := \boldsymbol{\theta} + \delta\boldsymbol{\theta}, \quad \delta\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \overline{\boldsymbol{\Sigma}}). \tag{5.11}$$

In addition, we utilize the generic measurement model

$$\mathbf{z}_i := \mathbf{h}_i(\boldsymbol{\theta}) + \mathbf{n}_i, \quad \mathbf{n}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_i). \tag{5.12}$$

Since most measurement types can be expressed in this common probabilistic form, other than the size of the vectors involved, there is no need to differentiate between measurement types.

The first step in the statistical test is to compute the estimated error for each measurement, $\overline{\mathbf{e}}_i$. This is defined as

$$\overline{\mathbf{e}}_i := \mathbf{z}_i - \mathbf{h}_i(\overline{\boldsymbol{\theta}}). \tag{5.13}$$

We term this the *estimated error* because the true state is unavailable, so the measurement error is *estimated* based on the hypothesis model. In the standard test that is utilized by RANSAC (Fischler and Bolles, 1981), the estimated error is assumed to follow the same probability distribution as the true error. That is,

$$\overline{\mathbf{e}}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_i). \tag{5.14}$$

However, this standard approach does not account for the uncertainty in the hypothesis model, which can lead to many incorrect classifications. Instead, propagating the hypothesis model uncertainty leads to

$$\overline{\mathbf{e}}_i \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}_i + \overline{\mathbf{H}}_i \, \overline{\boldsymbol{\Sigma}} \, \overline{\mathbf{H}}_i^T\right), \quad \overline{\mathbf{H}}_i := \left.\frac{\partial \mathbf{h}_i(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\overline{\boldsymbol{\theta}}}. \tag{5.15}$$

Using this distribution, we classify the measurements by computing a Mahalanobis distance for each estimated error. The distances associated to inlier measurements should follow a chi-squared distribution with a degree-of-freedom equal to the dimension of measurement space:

$$\bar{\mathbf{e}}_i^T \left(\mathbf{R}_i + \overline{\mathbf{H}}_i \overline{\boldsymbol{\Sigma}} \overline{\mathbf{H}}_i^T \right)^{-1} \bar{\mathbf{e}}_i \sim \chi^2_{\dim(\bar{\mathbf{e}}_i)}. \tag{5.16}$$

Therefore, measurements that exceed a user-defined threshold are classified as outliers, producing a set of measurement classifications. The linearized error models necessary for measurement classification in our mapping scenario are provided in (5.4) and (5.7).

Since we evaluate each measurement independently, the computation required scales linearly with the number of measurements. That is, given N measurements, the computational complexity of this outlier rejection algorithm is $O(N)$.

For robustness, this classification criterion is incorporated directly into the alignment algorithm through iterative reclassification. Since the batch alignment algorithm produces an improved estimate after each iteration, a classification step can be performed using the new hypothesis. The inlier measurements that pass the classification test remain in the alignment problem, while the ones that fail do not. The incorporation of iterative classification into the alignment algorithm is illustrated in Figure 5.8.

Finally, since the classification performance depends on the quality of the hypothesis model, we seek a robust initial estimate to seed the alignment algorithm. Based on the experiments presented in Tong and Barfoot (2013a), we utilize a multistage approach to achieve this requirement. This approach is summarized by the following steps:

1. If available, compute an initial dead reckoning estimate using the odometry measurements alone.
2. Perform pairwise RANSAC between all pairs of scans to provide a rough outlier classification.

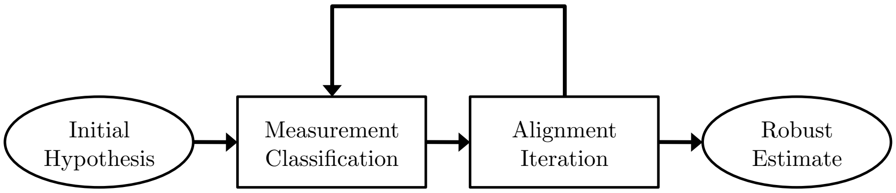


Figure 5.8: Iterative reclassification framework for robust estimation. By basing the classifications on the ever-improving hypothesis, classification accuracy will increase after each iteration.

3. Compute a batch alignment using the measurements classified as inliers by RANSAC.
4. Utilize the alignment as an initial guess to the batch alignment algorithm with iterative reclassification using the NIS test.

This multistage approach produces accurate classifications by first performing an initial rough classification using RANSAC, which detects the majority of the feature measurement outliers. Due to feature scarcity, it is computationally tractable to perform RANSAC between all pairs of scans. Though algorithmic efficiency can be improved by considering the spatial distribution of scans, this brute force approach ensures the best performance possible by avoiding any approximations. These classifications are then utilized to compute a batch alignment, producing an initial guess that can be refined using the NIS test within the iterative reclassification framework.

The presence of outliers further reinforces the need for a post-alignment verification check, since measurements are discarded by the outlier rejection tests. Though this may result in reduced coverage, it is better than incorporating outlier measurements.

5.7 Experiment Metrics

To evaluate the performance of the algorithm, we define a set of metrics to allow for quantitative analysis. Since we are considering a SLAM problem, the metrics consist of localization accuracy and map quality[2]. These metrics are similar to those proposed by Kümmerle et al. (2009), but all pose pairs are considered because we are interested in evaluating the global consistency of the solution. The map quality metric follows a similar theme, with the map error evaluated under the maximally consistent alignment.

5.7.1 Localization Accuracy

Though localization accuracy is less important than the overall map quality in the offline mapping scenario, it is typically reported as a SLAM performance metric. In light of this, we provide a measure of localization accuracy, but only compute it for the poses

[2] The metrics defined in this section are not the original equations presented in Tong et al. (2012a). The root mean squared (RMS) errors were originally defined to be the average error over the individual dimensions. This has been replaced by the more conventional RMS error definition that combines all of the translation components into a single Euclidean distance, and similarly, the rotation components. In effect, all of the original translation and rotation errors are increased by a factor of $\sqrt{3}$, and the affected computations and figures have been updated accordingly.

used to construct the map. Accurate localization is not required for all rover poses, since the exploratory traverse is already completed. To reiterate, this metric does not identify performance for local rover navigation, since the online localization scenario was not considered in this work.

Rather than computing the difference in alignment between the estimated and ground truth values in a single global reference frame, we utilize a relative measure instead. This is motivated by the fact that the SLAM problem is unobservable; there is an unknown translation/rotation with respect to a global reference frame. Since the measurements used by the ground truth system are independent from the measurements used in estimation, there are no shared reference points. Without a common frame shared between the ground truth system and the SLAM problem, it is difficult to justify the use of a single privileged frame for evaluating the estimation accuracy (Tong and Barfoot, 2011b).

We evaluate the position and orientation localization accuracy separately by defining the relative rover position estimate error, $\bar{\mathbf{e}}_{\boldsymbol{\rho}_{k_2,k_1}}$, and the relative rover orientation estimate error, $\bar{\mathbf{e}}_{\boldsymbol{\phi}_{k_2,k_1}}$, between poses k_1 and k_2 to be

$$\bar{\mathbf{e}}_{\boldsymbol{\rho}_{k_2,k_1}} := \boldsymbol{\rho}_{k_1}^{k_2,k_1} - \bar{\boldsymbol{\rho}}_{k_1}^{k_2,k_1}, \tag{5.17}$$

$$\bar{\mathbf{e}}_{\boldsymbol{\phi}_{k_2,k_1}}^{\times} := -\ln\left(\bar{\mathbf{C}}_{k_2,k_1}^{T}\mathbf{C}_{k_2,k_1}\right), \tag{5.18}$$

where $\bar{(\cdot)}$ denotes the estimated values, and we utilize the logarithmic map (4.10) for rotation matrices to obtain a rotation vector parametrization (Hughes, 1986). Using these expressions, we define our metric for localization accuracy to be the root mean squared (RMS) error of the relative transformations between all pairs of poses. Pose estimates that fail the automatic verification check are not included, since the estimates are deemed inappropriate for inclusion. Therefore, the localization accuracy metric for the rover pose translation and rotation estimates is expressed as

$$\hat{E}_{\text{trans}} := \sqrt{\frac{\sum_{k_1=1}^{K_v}\sum_{k_2=k_1+1}^{K_v}\bar{\mathbf{e}}_{\boldsymbol{\rho}_{k_2,k_1}}^{T}\bar{\mathbf{e}}_{\boldsymbol{\rho}_{k_2,k_1}}}{M_p}}, \tag{5.19}$$

$$\hat{E}_{\text{rot}} := \sqrt{\frac{\sum_{k_1=1}^{K_v}\sum_{k_2=k_1+1}^{K_v}\bar{\mathbf{e}}_{\mathbf{C}_{k_2,k_1}}^{T}\bar{\mathbf{e}}_{\mathbf{C}_{k_2,k_1}}}{M_p}}, \tag{5.20}$$

$$M_p := \binom{K_v}{2},$$

where K_v is the number of poses deemed valid by the automatic verification check. This metric identifies overall localization consistency, rather than alignment accuracy to an unknown global frame.

5.7.2 Map Quality

The map quality can be evaluated by computing the average point mismatch between the estimated and reference dense maps. Since there is no shared common frame, we compute the map quality by first seeking the best alignment between the maps. This is accomplished by initializing with the ground truth pose of the estimated map, and computing the optimal adjustment using ICP (Besl and McKay, 1992). With the optimal alignment, the map error for an estimated point, $\bar{\mathbf{q}}_i$, is computed by subtracting its position from the position of its nearest neighbour in the dense reference map, $\mathbf{q}_{nn_i}$:

$$\bar{\mathbf{e}}_{\mathbf{q}_i} := \mathbf{q}_{nn_i} - \bar{\mathbf{q}}_i. \tag{5.21}$$

Using this expression, we define the map accuracy metric as the RMS error over all dimensions and all the points in the estimated map. This is expressed as

$$\hat{E}_{\text{map}} := \sqrt{\frac{\sum_{i=1}^{N_p} \bar{\mathbf{e}}_{\mathbf{q}_i}^T \bar{\mathbf{e}}_{\mathbf{q}_i}}{N_p}}, \tag{5.22}$$

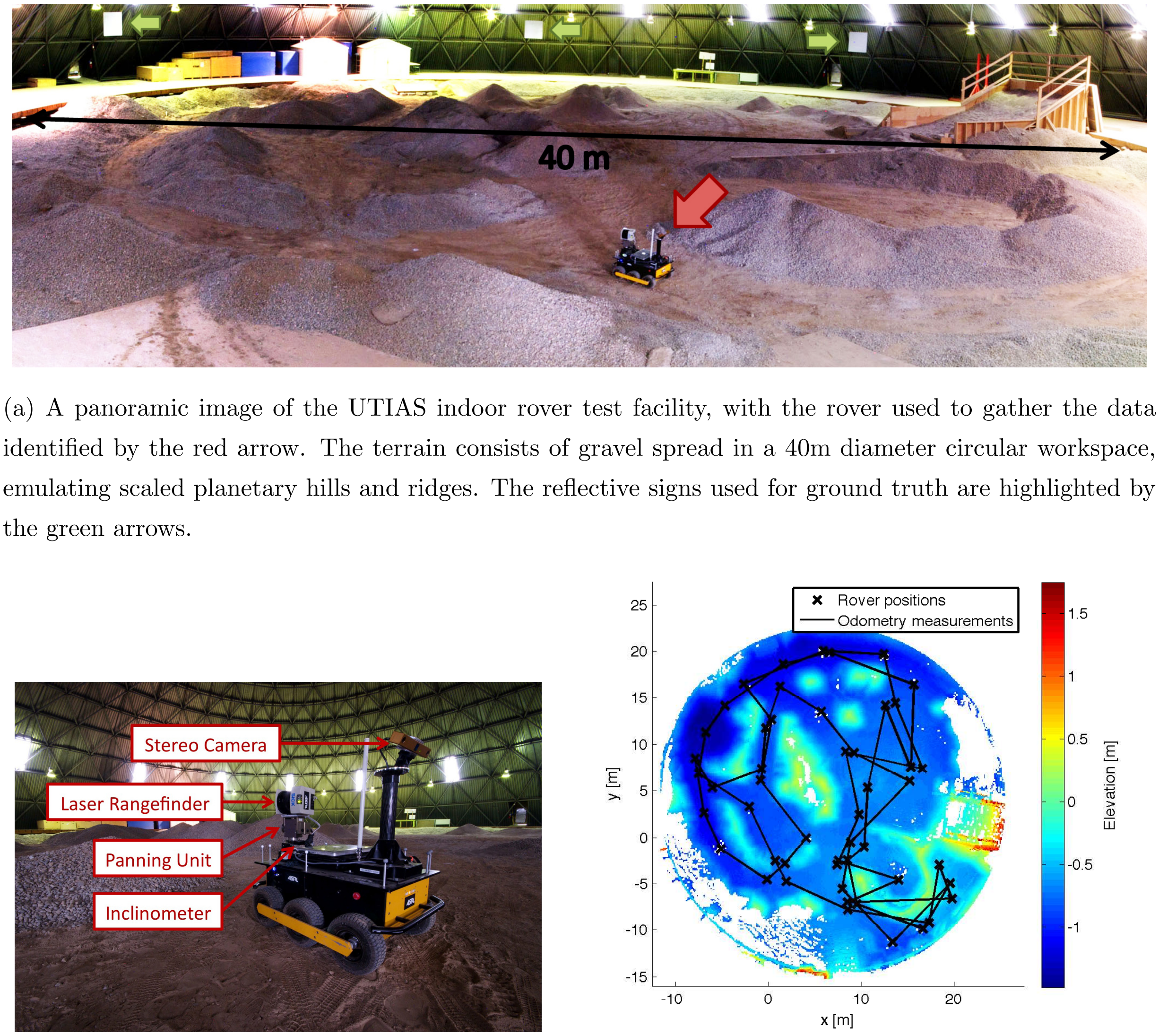
where N_p is the number of points in the estimated dense map. This approach identifies the map quality under the maximally consistent alignment between maps.

5.8 UTIAS Indoor Rover Test Facility

Validation of the proposed framework was conducted using scans obtained from two planetary analogue test facilities. By utilizing data from two separate sites, we were able to demonstrate the robustness of the approach to different environmental conditions. To reflect the conventional approach for real-world algorithm development, a subset of the data from the first site was used as a training set to tune the algorithm parameters. These parameters remained the same for the second site, which emulated a deployment to an unknown location. The specific values of the tuning parameters used during the experiments are listed in Tong et al. (2012a), and the experimental data used to test the proposed framework have been made available in Tong et al. (2013c).

5.8.1 Overview

The first set of data was obtained at the University of Toronto Institute of Aerospace Studies (UTIAS) indoor rover test facility. The facility consists of a large dome structure, which covers a workspace area 40m in diameter. In this workspace, gravel was



(a) A panoramic image of the UTIAS indoor rover test facility, with the rover used to gather the data identified by the red arrow. The terrain consists of gravel spread in a 40m diameter circular workspace, emulating scaled planetary hills and ridges. The reflective signs used for ground truth are highlighted by the green arrows.

(b) The Clearpath Husky A100 rover used to gather the data at UTIAS. The payloads consist of a laser rangefinder mounted on a panning unit to provide the 3D laser scans, an inclinometer for pitch and roll correction, and a stereo camera for visual odometry.

(c) A plot depicting an overhead view of the reference map of the dome and the rover scan locations. To put this plot into the perspective of (a), the panoramic image was taken looking-in from the bottom of the plot.

Figure 5.9: The hardware and test site used for the UTIAS indoor rover test facility dataset, as well as the distribution of scans obtained in the terrain.

distributed to emulate scaled planetary hills and ridges, providing characteristic natural, unstructured terrain. An image of the terrain is provided in Figure 5.9(a).

The rover used to gather the dataset, depicted in Figure 5.9(b), carried a number of payloads. The relevant payloads consisted of a laser rangefinder mounted on a panning unit, an inclinometer, and a stereo camera. The rover was deployed in a looping traverse, driven to explore the terrain in an exploratory manner. Periodic stops were conducted to pan the laser rangefinder, obtaining full 360×180° scans of the local environment. In addition, stereo image sequences were obtained between scans, capturing images of the terrain while the rover was in motion. A total of 50 scans were taken over the course of three loops around the terrain. The scan locations and the associated odometry measurement links are depicted in Figure 5.9(c).

The ground truth of the rover was provided by the identification of known landmarks outside the workspace. These consisted of large retroreflective signs mounted on the dome structure, which was able to provide six degree-of-freedom (6DoF) poses with centimetre-level accuracy in translation, and half-degree accuracy in orientation (Tong and Barfoot, 2011b). The reference map was produced using the ground truth poses as initial guesses, and an ICP algorithm for fine adjustment. We utilized a metascan approach (Chen and Medioni, 1992), where scans were combined after pairwise ICP alignment to progressively build up a complete map. This approach maintained global consistency, and overcame the issue of low scan overlap.

Since the workspace lay in the middle of an artificial dome structure, the scans were trimmed to remove the points that fell outside of the workspace. In addition, to reflect the dimensions of a full-scale worksite terrain, the maximum range of the sensor was artificially limited to 10m. The pitch and roll corrections for the feature detection step were provided by the onboard inclinometer, and the stereo image sequences were utilized for VO.

Since the camera range could not be easily scaled in the same manner through range-limiting, we emulated the effects of a longer planetary worksite traverse by utilizing a typical implementation of VO (Furgale and Barfoot, 2010) under poor lighting conditions. This provided motion blur, which reduced the accuracy of the motion estimates. Preliminary tests conducted using a motion tracking system demonstrated a 7% average error of distance travelled, over a short 70m traverse. This ensured that we could not produce a high-quality map using the VO estimates alone, which emulates an actual planetary worksite mapping scenario.

5.8.2 Results

In this section, we present the results of the mapping framework when applied to the dataset. The 50 scans were utilized in two ways for validation. In the first section, the robustness of the framework is demonstrated by utilizing subsets of the data to produce a large set of trials. With this quantitative statistical analysis, a more qualitative sense of the algorithm is then provided by utilizing all of the scans to produce a single, accurate map. This is accompanied by discussions on the limitations of the framework, as well as rendered scenes for visual validation.

To provide perspective, three approaches to mapping are compared:

(i) Dead reckoning, using the odometry measurements alone.
(ii) Our framework, without the post-alignment automatic verification check.
(iii) Our framework, with the post-alignment automatic verification check.

The comparison of these three approaches provides perspective on the magnitude of errors, as well as highlights the importance of an automatic verification check prior to final map construction.

5.8.2.1 Subsets

To test the robustness of the framework, a large set of trials were generated by utilizing subsets of the scan data. For each of the trials, 20 scans were chosen from the full set of 50. If consecutive scans were chosen, the odometry measurements were included, otherwise, they were not. Without any additional knowledge, the dead reckoning estimate handled gaps in the odometry by assuming no motion with a very large uncertainty.

These subsets were constructed by regularly selecting points in the workspace, and collecting the 20 nearest scans. This method ensured spatial coherence, and encouraged sufficient overlap between scans to produce a map estimate. Spatial coherence is necessary since non-overlapping scans would provide an impossible alignment problem to solve. In total, 576 unique subsets were constructed. Though the majority of the trials produced good alignments, 2% of the cases resulted in an average position alignment error of more than 1m. This can be attributed to an insufficient amount of data to correctly identify outlier measurements, which can result in a poor estimate. Further discussion of the failure modes of the proposed framework is provided in Section 5.10.1. Table 5.1 summarizes the nominal alignment performance for the remaining 565 trials.

Table 5.1: Average RMS quality metrics for the nominal trials of the UTIAS indoor rover test facility dataset.

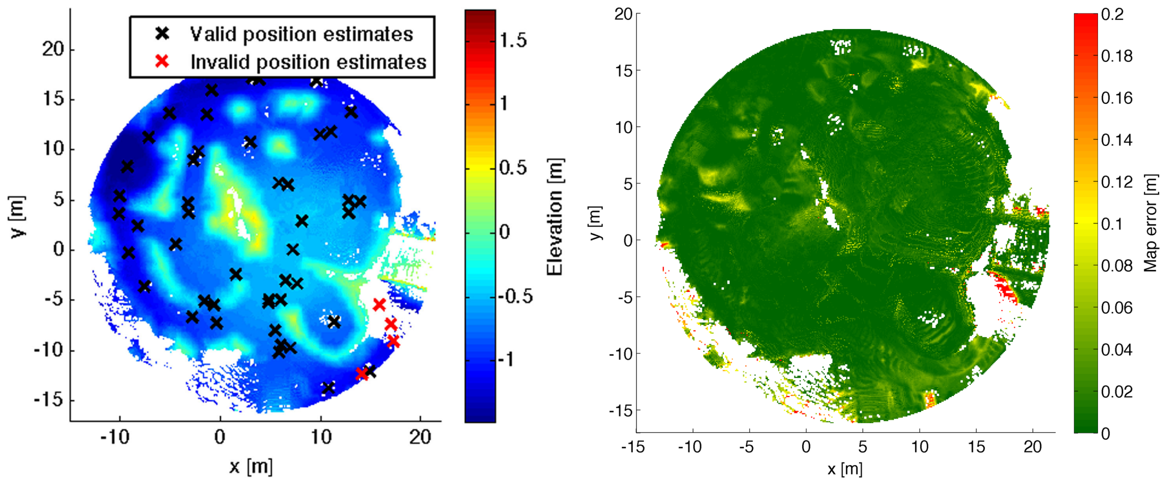
	$\hat{E}_{\text{trans}}$		$\hat{E}_{\text{rot}}$		$\hat{E}_{\text{map}}$	
	Position Error [m]		Orientation Error [°]		Map Error [m]	
	Mean	Std Dev	Mean	Std Dev	Mean	Std Dev
(i) Dead reckoning	17.7	3.93	100.3	9.72	3.34	2.55
(ii) Framework without check	2.84	3.53	16.7	21.7	0.64	0.80
(iii) Framework with check	1.05	0.24	3.65	1.73	0.12	0.05

As can be seen, while the proposed framework offered a large improvement over simple dead reckoning, the post-alignment automatic verification check was a vital component to producing an accurate map. On average, 19 out of the 20 scans were considered valid for inclusion in each alignment trial. While our metrics value accuracy and do not apply a penalty for lack of coverage, the good performance over a large set of trials illustrate the robustness of the framework.

5.8.2.2 Complete Dataset

As suggested by the performance in the robustness trials, the application of the framework to all 50 scans produced results with similar levels of accuracy. Of the 50 scans, 46 estimated poses were considered valid for inclusion in the final map estimate, which is depicted in Figure 5.10(a). As can be seen, all four of the invalid scans were located behind the ridge structure depicted on the right side of Figure 5.9(a). Due to the height of the ridge, the sensor range was severely limited. This resulted in a lack of common features between scans on either side of the ridge.

Though odometry measurements provided a link to these poses, the long distances travelled produced large uncertainties. The combination of a lack of certain measurements and the corruptive effects of outlier measurements led to estimates that were highly uncertain. As a result, the pose estimates were deemed inappropriate for the final map. For the remaining 46 valid poses, the metric values were 1.53m in position, 4.21° in orientation, and 0.07m in the map. In addition, we recorded timing information for our proof-of-concept implementation to provide a sense of how the processing time was



(a) Estimated map and rover poses.

(b) Map error.

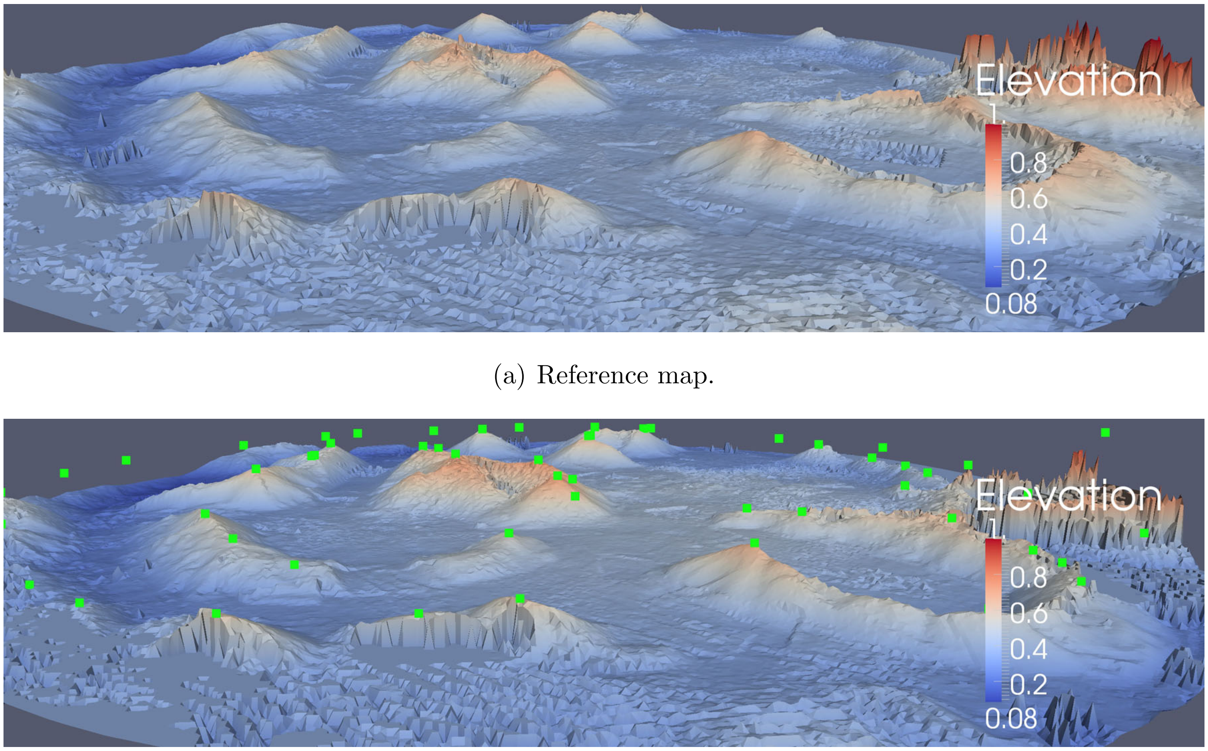
Figure 5.10: Plots depicting overhead views of (a) the estimated map and (b) the mapping error. In (a), the majority of the estimated poses were considered valid for inclusion into the final map, except for the four behind the ridge structure indicated by the red Xs. This can be attributed to a lack of common features due to occlusions. This is visually confirmed in (b), where the large errors are confined to the regions with little scan overlap.

divided. In total, the run time for the complete dataset was 1.3h, which was divided into 15mins for feature extraction, 1h for feature matching, and 5mins for batch alignment[3].

Though the average metric values provide a good indication of the overall performance of the framework, it does not indicate where the mapping errors arose. Since the goal of worksite mapping is to create an accurate map, additional analysis was conducted to evaluate the map quality. In Figure 5.10(b), we provide a plot of the map errors, $\hat{E}_{\text{map}}$, computed for each individual point. As can be seen, the majority of the map was very accurate. The larger errors are confined to the area behind the ridge structure and near the peaks of the terrain, where there was little overlap between scans due to occlusions.

Finally, to obtain a qualitative sense of the map accuracy, meshes were created to provide visual validation. In Figure 5.11, we have rendered the reference and estimated maps from a similar perspective as Figure 5.9(a). As can be seen, the estimated map visually resembles the reference map. Some small deviations can be seen at the peaks and behind the ridge structure, as indicated by Figure 5.10(b).

[3] Timing information recorded on a MacBook Pro with a 2.66GHz Core 2 Duo and 4GB of 1067MHz DDR3 RAM in Matlab, utilizing both cores.



(a) Reference map.

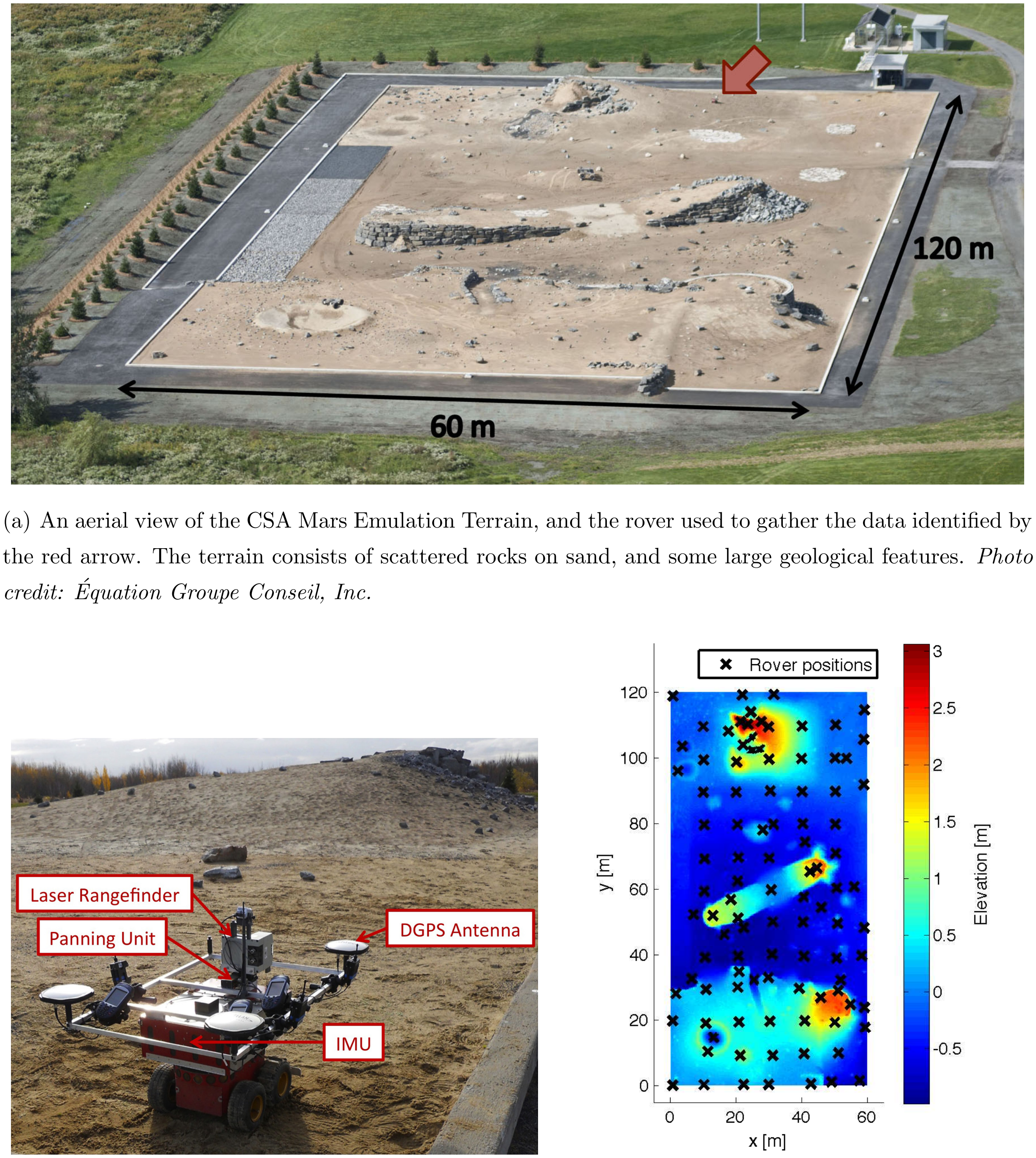
(b) Estimated map.

Figure 5.11: Meshes created from the reference and estimated point clouds, rendered from a similar perspective as Figure 5.9(a). In (b), the green squares mark the estimated sparse feature positions. As can be seen, the two maps are visually similar, with small deviations near the peaks of the terrain.

5.9 CSA Mars Emulation Terrain

5.9.1 Overview

The second set of data was obtained at the Canadian Space Agency's (CSA) Mars Emulation Terrain, depicted in Figure 5.12(a). The outdoor test facility has dimensions of 120m x 60m, and contains ridges, craters, and outcrops. By considering this substantially larger site as a proving ground for the framework, its performance should illustrate its validity for planetary mapping. To reiterate, the tuning parameters for the framework were determined using a subset of the data from the first site, and remained the same for the second site. This emulated a deployment to an unknown location. Furthermore, the dataset was collected by an independent party using a different rover. This approach avoided any unintentional bias in data collection, and resulted in a challenging dataset.



(a) An aerial view of the CSA Mars Emulation Terrain, and the rover used to gather the data identified by the red arrow. The terrain consists of scattered rocks on sand, and some large geological features. *Photo credit: Équation Groupe Conseil, Inc.*

(b) The modified P2AT rover used at CSA. The 3D laser scans were obtained using a laser rangefinder mounted on a panning unit, inclination by an internal IMU, and ground truth positions and yaw by the DGPS antennas.

(c) A plot depicting an overhead view of the reference map of the terrain and the rover scan locations. This is rendered from the same direction as Figure 5.12(a), but at an overhead view.

Figure 5.12: The hardware and test site used for the CSA Mars Emulation Terrain dataset, as well as the distribution of scans obtained in the terrain.

The rover used to obtain the dataset, depicted in Figure 5.12(b), also carried a laser rangefinder mounted on a panning unit, and an IMU to measure inclination. Scans were obtained with an approximately regular spacing of 10m, producing a total of 102 scans of the terrain. The ground truth poses were obtained using an array of three DGPS antennas, which provided position and yaw. Unfortunately, due to the small baseline of the antenna array, accurate pitch and roll measurements could not be obtained. Since it would be inappropriate to utilize the same sensor measurements for estimation as well as metric evaluation, the IMU measurements were only used to provide rotations for reference map construction, but not for the localization accuracy metric. The reference map was produced using the ground truth poses for alignment. For these experiments, no forms of odometry were recorded.

Due to the size of the terrain, the maximum effective range of the sensor was retained at 30m. This ensured that sufficient overlap for matching would remain between scans. Since odometry measurements were unavailable, no pose-to-pose transformation estimates were used in the alignment algorithm. In effect, the batch alignment algorithm was initialized by placing all of the scans at the same location, with a very large uncertainty. Though this approach may result in alignment failure due to feature scarcity, limitations in the data must be accepted. The scan locations are depicted in Figure 5.12(c).

5.9.2 Results

Similar to the first dataset, the second set of data was also utilized in two ways. Robustness is first demonstrated by using subsets of the data, and then all of the scans are utilized to produce a single map for qualitative analysis. For this dataset, we computed a modified form of the localization accuracy metric for the rotation estimates. Since only ground truth yaw was available, we replaced the rotation accuracy metric, $\hat{E}_{\text{rot}}$, with a yaw accuracy metric, $\hat{E}_{\text{yaw}}$. The computation of this metric was based upon (5.20), but modified to account for yaw only.

5.9.2.1 Subsets

The trials for the robustness tests using the Mars Emulation Terrain data were generated in the same way as the dome trials. In total, 1507 unique subsets of 20 scans were produced from the full set of 102 scans. Though spatial coherence was encouraged, insuf-

Table 5.2: Average RMS quality metrics for the nominal trials of the CSA Mars Emulation Terrain dataset.

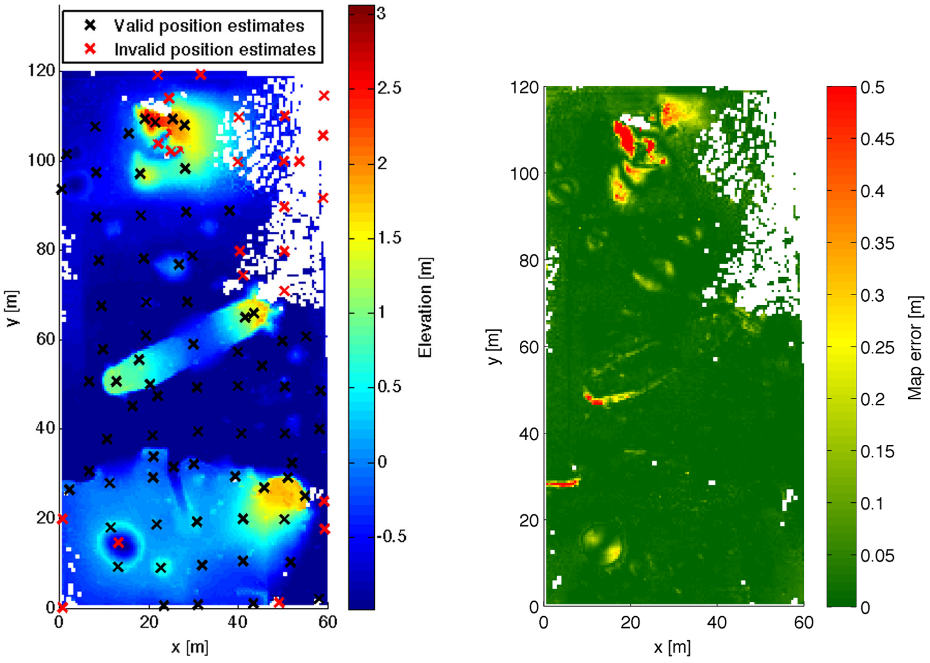
	$\hat{E}_{\text{trans}}$		$\hat{E}_{\text{yaw}}$		$\hat{E}_{\text{map}}$	
	Position Error [m]		Yaw Error [°]		Map Error [m]	
	Mean	Std Dev	Mean	Std Dev	Mean	Std Dev
(i) No motion	22.9	2.37	103	4.61	0.66	0.17
(ii) Framework without check	10.7	8.24	34.1	28.6	2.34	1.73
(iii) Framework with check	0.57	0.21	1.27	0.55	0.26	0.17

ficient measurements led to off-nominal behaviour in 0.6% of the trials. These behaviours included rejection of all poses in the post-alignment check, and large position estimate errors due to the presence of outlier measurements. Since both behaviours would skew the average metrics, the off-nominal trials were not included. Table 5.2 summarizes the nominal alignment performance for the remaining 1497 trials.

As expected, the initial estimate of no motion, where all of the scans are situated in the same location, produced the largest errors. These errors were only limited by the fact that the subsets were chosen to be spatially close. Once again, while the proposed framework provided some improvement, the post-alignment automatic verification check was a vital component to producing an accurate map. On average, 15 scans of the 20 were classified as valid for each of the trials. This lower number can be attributed to the feature scarcity and large scan spacing.

5.9.2.2 Complete Dataset

When the framework was applied to the complete dataset, the alignment estimate resulted in 75 of the 102 scans classified as valid for inclusion into the final map. This map is depicted in Figure 5.13(a). These invalid poses were localized to a few regions that suffer from feature scarcity. In particular, the majority of the invalid poses were obtained in the flat region depicted in the upper-right corner of Figure 5.12(a). In this region, the few distinct rocks provided an insufficient number of feature matches to produce a confident alignment estimate. For the remaining valid poses, the metric values were 1.63m in position, 1.47° in yaw, and 0.16m in the map. Though the position error appears to be



(a) Estimated map and rover poses.

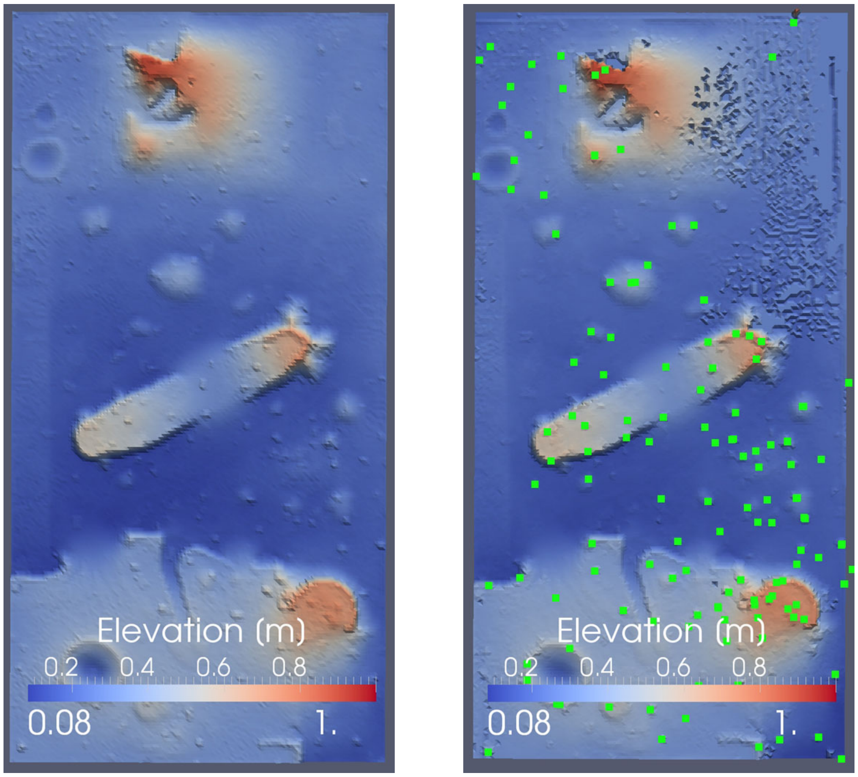
(b) Map error.

Figure 5.13: Plots depicting overhead views of (a) the estimated map and (b) the map accuracy. In (a), the invalid poses, indicated by the red Xs, are localized to a few regions. The upper-right corner is unmapped due to insufficient distinct terrain features in the flat region to produce a confident estimate. Other invalid poses are at the edges of the terrain, or in low regions where occlusions dominate much of the scan. For (b), the large errors are once again confined to the peaks, where there is little scan overlap.

larger compared to the robustness trials, it should be noted that the scale of the complete dataset was much larger than each of the subsets. The total run time for the complete dataset was 12.5h, which was divided into 0.5h for feature extraction, 10.5h for feature matching, and 1.5h for batch alignment[4].

In Figure 5.13(b), we provide a plot of the map errors. Similar to mapping the indoor rover test facility, the major errors were experienced at the high points of the terrain, where there was little scan overlap. The upper-right corner is left unmapped due to the post-alignment automatic verification check, but overall, the estimated map appears to be quite accurate. This is visually confirmed by the overhead renders depicted in Figure 5.14. The overall structure has been accurately captured, with small deviations appearing at the edges of the terrain features, and in the unmapped upper-right corner.

[4] Timing information recorded using the same computer as the UTIAS dataset.



(a) Reference map.

(b) Estimated map.

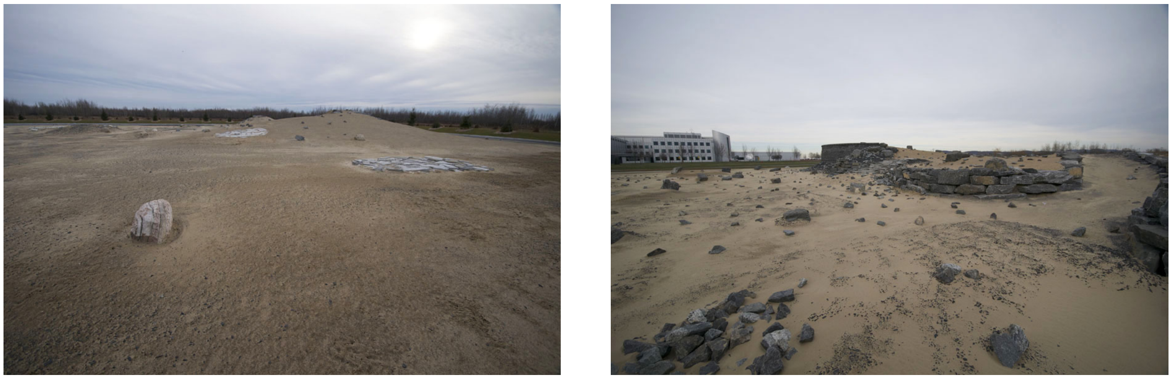
Figure 5.14: Meshes created from the reference and estimated point clouds, rendered from an overhead view, as well as the sparse feature positions marked by the green squares in (b). The two maps are visually similar, with small deviations appearing at the edges of terrain features, and the unmapped upper-right corner.

5.10 Discussion

Although the overall performance of the framework was positive, some shortcomings warrant further analysis. In particular, we address the percentage of unused scans due to insufficient feature matches, and the few major localization errors experienced in both sets of robustness trials. This is presented through analysis of the failure modes of the current implementation, and possibilities for future improvements. A detailed comparison between the dense and sparse feature approaches can be found in Tong et al. (2012a).

5.10.1 Failure Modes

Though the framework was tested using data from two locations, the examples provided in this section will only refer to mapping the CSA Mars Emulation Terrain, because the



(a) Feature-poor flat region.

(b) Feature-rich middle region.

Figure 5.15: Two images depicting ground-level views of the CSA Mars Emulation Terrain. In (a), the flat region located in the upper-right corner of Figure 5.12(a) is relatively feature-poor, with only the rock in the foreground able to serve as a distinctive geometric feature. This is in stark contrast to (b), which was taken in the middle region.

results better highlight the shortcomings of the framework. Since odometry was unavailable, the only measurements used for alignment were the distinctive peaks identified in the scans. Therefore, the primary cause of failures of the framework stemmed from the feature extraction and mapping stage.

When utilizing the complete dataset, 27 of the 102 estimated poses and their associated scans were deemed unsuitable for inclusion into the final map. For many of these poses, too few feature matches led to an insufficient number of measurements to produce a confident pose estimate. For example, Figure 5.15(a) provides a ground-level view of the flat region of the terrain in the upper-right corner of Figure 5.12(a). This is a stark contrast to the relatively feature-rich region from the middle of the terrain, depicted in Figure 5.15(b), where the alignment algorithm performed very well. Apart from the solitary rock in the foreground, there is little distinctive geometric information suitable for identification and matching. This leads to a weak network of feature matches, which is very sensitive to measurement noise. Experiments conducted by the CSA have shown that a dense matching approach performs better in this region (Gemme et al., 2012).

Although improved mapping performance could result by increasing the scan overlap in the feature-poor regions, an important point to highlight is the fact that many of these failures at the feature matching stage were appropriately detected by the post-alignment automatic verification check. If we compare against ground truth, only 2 of the 27 scans classified as unsuitable for inclusion were incorrectly identified. Similarly, a single estimated pose was misclassified as suitable for inclusion by the verification check. However, based on the generated plots, these errors did not seem to have a large effect on the quality of the resulting map. Both types of incorrect classifications can be attributed to the fact that we use a threshold on the estimated uncertainty values for the verification check. When estimation is conducted with respect to a single reference frame, even correctly aligned distant poses are simply very uncertain. Therefore, it is difficult to distinguish between good and bad alignments at a long range. One possible modification to address this issue would be to repeat the estimation problem with different estimate reference frames, and compare the results. This post-alignment check introduced additional redundancy into the framework, which is essential for robust real-world operation.

In addition, we must be aware of the fact that incorrect matches are an inevitable part of an automatic system. Ambiguities may arise due to regions similar in appearance. As a result, the inclusion of outlier rejection into the framework is essential for robust operation. However, since the other measurements are used to identify outliers, it is possible that a scenario arises where there are insufficient inlier data to identify outliers. It is important to note that the algorithm performance is fundamentally limited by the quality of the measurement data.

5.10.2 Possible Improvements

Since the failure modes experienced in the experiments were largely due to inadequate performance by the feature extraction and matching algorithm, improvements at that stage should have a significant impact on the overall performance of the framework. In particular, discussion is provided in this section on alternative feature extraction and matching methods, an area for computational savings, and an added component which should improve the overall map quality performance.

While peaks were chosen for their distinctiveness and simplicity in planetary worksite terrain, alternative features can be identified in flat terrain by replacing the detector. Other possibilities include regions of high curvature (Bakambu et al., 2006b), or geometric rock models (Li et al., 2007). In addition, the dense data could be utilized in select scenarios where feature extraction fails. These supplementary methods could be triggered by the batch alignment algorithm, where the intermediate alignment estimates could indicate regions where further investigation is needed.

An additional heuristic applied to data association could also help reduce the computational requirements. As evidenced by the timing results, a large majority of the processing time is devoted to feature matching. In the current implementation, feature matching is conducted between all pairs of scans. Computational savings could be gained by simply limiting the matching to scans that are spatially close, which should significantly reduce the number of pairs to be tested. An estimate of locality could be obtained by using the intermediate alignment estimates.

Algorithmic improvements for data association could include utilizing Joint Compatibility Branch and Bound (JCBB) (Neira and Tardós, 2001) instead of DARCES alone, which should reduce the number of similar triangle hypotheses through batch gating. Alternatively, feature extraction and matching can be sped up by limiting the search space to consistent hypotheses. For example, the feature position estimate from one scan could provide an indication of where the feature should appear in an another scan. This would be an adaptation of the Active Matching algorithm (Chli and Davison, 2009).

Finally, as mentioned previously, the map accuracy could be improved by reintroducing the dense data at the final map construction stage. Though the feature-based approach provides computational savings by encapsulating the terrain data into point landmarks, the fine information necessary for exact alignment is lost. Therefore, the proposed framework could be used to provide an initial guess, bringing the scans to the correct attractor basin for a dense matching method. Expected scan overlap can be computed to ensure appropriate performance through dense alignment, and global consistency could be maintained using a global relaxation method (Borrmann et al., 2008).

5.11 Conclusion

In summary, we have presented a complete 3D SLAM framework suitable for autonomous mapping of a planetary worksite environment. Operation in a planetary environment imposes sensing restrictions, as well as significant challenges due to the rugged terrain. By utilizing a panning laser rangefinder mounted on a rover platform, we have demonstrated an approach that is able to create globally consistent maps of natural, unstructured terrain.

To accomplish this, we extract distinct peaks identified in the laser scans, and match them using a combination of sparse-feature constellations and dense data. These feature measurements are then utilized in combination with pose-to-pose odometry transformation in a batch alignment algorithm to provide a globally consistent estimate. For robustness, the alignment algorithm is also augmented with heterogeneous outlier rejection to address the possibility of outliers in either measurement type. Finally, a post-alignment automatic verification check is performed to ensure that the estimates produced by the alignment algorithm are suitable for inclusion into the final map.

For validation, the framework was tested using experimental data collected from two different planetary analogue test facilities, and using two different rovers. For each site, the data was split into subsets to evaluate the robustness of the algorithms, as well as utilized as a whole to construct an estimated map of the entire site. Quantitative metrics were provided for localization and map accuracy, as well as qualitative analysis through visual validation. In addition, detailed analysis was provided on the failure modes of the framework, as well as possible improvements for future iterations. Overall, the accurate mapping results strongly validates the effectiveness of our framework.

To summarize, while some novelty can be found in the re-formulations of the algorithms adapted to our scenario, the major contributions of this chapter are:

1. The presentation of the first laser-based 3D SLAM algorithm that incorporates hybrid data association, batch alignment, heterogeneous outlier rejection, and a post-alignment automatic verification step into a complete, working system suitable for robust, autonomous mapping of a natural, unstructured worksite environment.
2. Large-scale validation of the proposed framework using data from two different planetary analogue facilities, including 6DoF ground truth and detailed analysis on the successes and failure modes.

There are a number of possible avenues for future work. While these may include the improvements discussed in the previous section, an important point to consider is that mapping is only one aspect of the site surveying task. Our mapping framework considers the problem of aligning the data after it has been collected, but navigation and exploration is required to collect it in the first place. In our experiments, we utilized stereo VO to obtain pose-to-pose estimates. Since passive cameras are sensitive to ambient light, we should be better served by using the laser rangefinder to obtain VO estimates as well. We address this problem in the next chapter.

Chapter 6

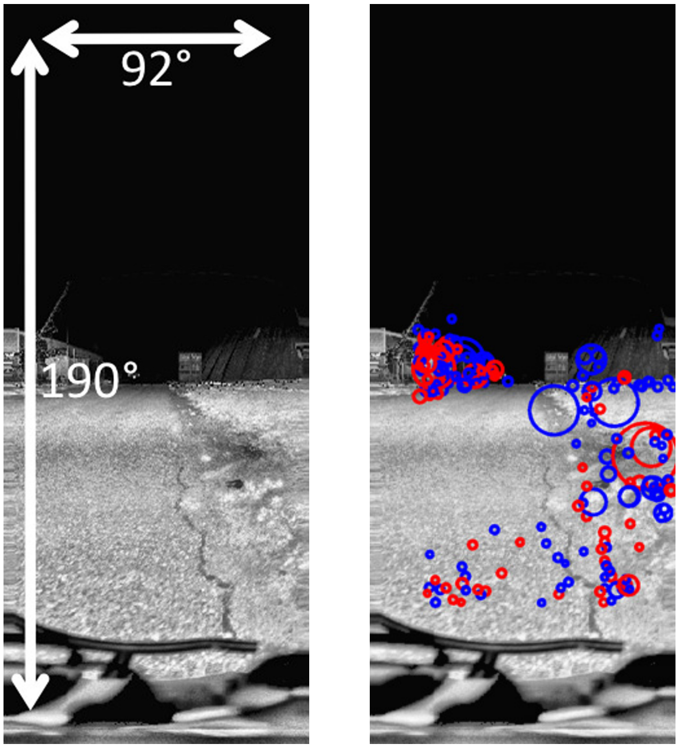
Laser-Based Visual Odometry

6.1 Introduction

In this chapter, we present a framework for obtaining Visual Odometry (VO) estimates using a panning laser rangefinder. This framework makes use of the continuous-time batch SLAM theory presented in Chapter 3, and the 3D state parametrizations provided in Chapter 4. Furthermore, it can serve as the motion estimation method necessary for navigation between scan stops in the planetary worksite mapping scenario.

For inspiration, we begin by looking towards existing VO methods in the current literature. In mobile robotics, the dominant form of motion estimation uses sparse, appearance-based techniques applied to stereo camera imagery. These methods offer robustness and accuracy, at reasonable computational costs. Stereo camera VO is employed for space exploration (Maimone et al., 2007; Johnson et al., 2008), and has demonstrated high accuracies over long distances (Konolige et al., 2007; Sibley et al., 2010b). We intend to leverage this prior work by transferring the core concepts to laser data.

Sparse appearance-based VO methods identify and track distinctive points in the camera imagery. Commonly used interest points include Harris corners (Harris and Stephens, 1988), Scale Invariant Feature Transforms (SIFT) (Lowe, 2004), and Speeded Up Robust Features (SURF) (Bay et al., 2008). These features are then input into a state estimation algorithm to determine the motion estimate that best explains the feature tracks. Recent work (McManus et al., 2011, 2012) has shown that images constructed from the laser intensity data resemble greyscale camera imagery, and can be utilized in the same manner in sparse appearance-based VO algorithms. This is visually confirmed by Figure 6.1, where we show a laser intensity image, and detected SURF features.



(a) Intensity image with contrast adjustments.

(b) Intensity image with detected SURF features.

Figure 6.1: Sample images constructed from panning laser intensity data that can be utilized for VO. Since these resemble greyscale camera images, we take an appearance-based approach where sparse SURF features are identified and tracked in consecutive frames. The SURF detector identifies distinct blobs in the image, which are displayed in red for dark blobs on a light background, and blue for light blobs on a dark background.

However, a key distinction between camera and laser sensors is the method of data acquisition. A camera sensor acquires an entire image at a single time instant, while a scanning laser rangefinder measures points individually. As a result, intensity images constructed while the rover is in motion exhibit distortion effects. Though a stop-scan-go approach may be employed to avoid the motion distortion issues (Nüchter et al., 2007; McManus et al., 2011), this mode of operation severely reduces the capabilities of the autonomous platform. Inertial measurement units (IMUs) can be used for motion correction, but this solution is undesirable due to the reliance on an additional sensor. Furthermore, calibration between multiple sensor types is very challenging (Kelly and Sukhatme, 2011). If a motion estimate can be obtained using the laser data alone, the algorithm can benefit when IMU data is available, and still operate when it is not.

We address the issue of motion distortion by considering the timestamps of the interest points detected in each image. Since each feature is detected at a different time instant, we formulate the state estimation problem in continuous time. This is facilitated by GPGN, which was introduced in Chapter 3. We model the rover pose as a GP, and employ GP interpolation to obtain the poses at the measurement times.

In summary, we consider the 3D laser-based VO problem in this chapter. We construct intensity images, and track sparse visual features for motion estimation. The effects of motion distortion are handled by conducting continuous-time state estimation using GPGN, and experimental validation is presented using 365m of outdoor data gathered by a panning laser rangefinder mounted on a field rover. This is the same laser sensor as used for mapping in the previous chapter, but configured to gather data continuously. Furthermore, we demonstrate that the VO framework is applicable to the planetary worksite mapping scenario by integrating the pose-to-pose VO estimates with the mapping framework to construct a map of planetary analogue terrain.

This chapter presents a practical application of the theoretical contributions introduced in Chapter 3. Portions of this chapter have appeared in Tong and Barfoot (2013b), where we demonstrated laser-based VO for a 1.1km traverse using a different high-framerate scanning lidar. However, keeping with the planetary worksite mapping theme of this thesis, we focus on using a slow-framerate panning laser rangefinder in this chapter.

The remainder of this chapter is organized as follows. We begin with a review of related work in Section 6.2. This is followed by an architectural overview of the VO pipeline in Section 6.3, detailed descriptions of the pipeline components in Sections 6.4 and 6.5, and experimental validation in Section 6.6. Finally, a discussion is provided in Section 6.7, and concluding remarks are made in Section 6.8.

6.2 Related Work

The use of laser intensity data in mobile robotics is not a new concept. Intensity images have been used for indoor localization against a known map (Neira et al., 1999), identification of retroreflective markers in the scene (Guivant et al., 2000; Tong and Barfoot, 2011b), and road localization for self-driving cars (Levinson, 2011).

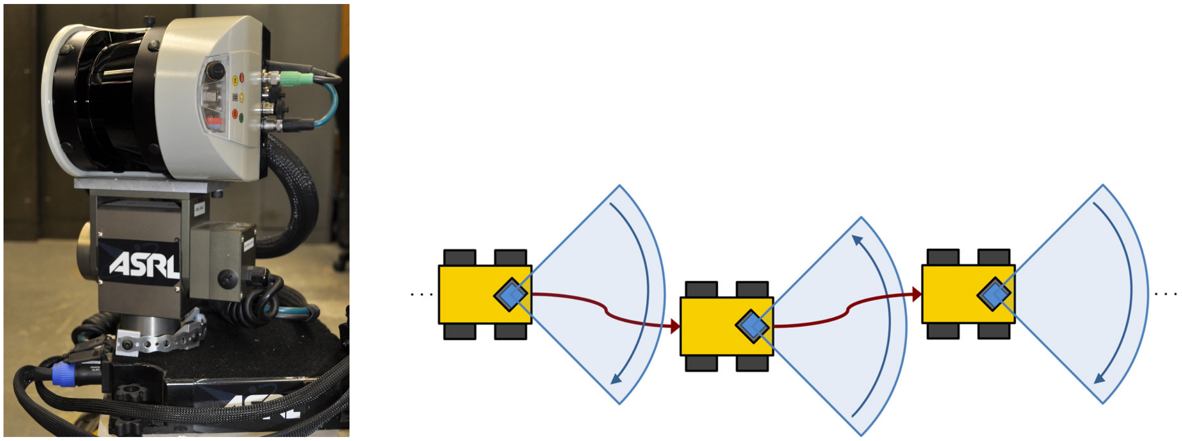
Previous work in laser motion estimation includes the SLAM algorithm tailored for a high-framerate Velodyne lidar developed by Moosmann and Stiller (2011), which performed a combination of dense matching using the Iterative Closest Point (ICP) (Besl and McKay, 1992) algorithm, and linear de-warping for motion correction. More generally, the continuously spinning laser VO work by Bosse and Zlot (2009) subdivided the dense data into discrete segments, and iteratively corrected each point cloud segment as the motion estimate was improved. These concepts were recently applied towards large-scale underground mapping (Zlot and Bosse, 2012), where interpolation was introduced to smooth the trajectory estimate, and an IMU was included to assist in motion compensation. A similar formulation is utilized in the *Zebedee* mobile mapping system (Bosse et al., 2012).

Appearance-based methods for laser data have been demonstrated to provide comparable accuracy to stereo camera VO for stop-scan-go operation (McManus et al., 2011), and sufficient for Visual Teach and Repeat (VT&R) even without motion compensation (McManus et al., 2012). Improved estimates were achieved by Dong and Barfoot (2012), who considered the motion distortion issues by employing a novel pose interpolation scheme to obtain the intermediate poses between successive frames. Similarly, Hedborg et al. (2012) performed Bundle Adjustment using rolling shutter camera imagery by employing the SLERP (Shoemake, 1985) quaternion interpolation method.

Further improvements can be achieved by considering the estimation problem in continuous time. Rather than simply performing interpolation between frames, a richer state representation can be chosen that better reflects the rover motion. Though continuous-time filtering methods predate much of mobile robotics (Kalman and Bucy, 1961; Jazwinski, 1970), it is only recently that continuous-time batch state estimation algorithms have appeared in the literature. These include using a piecewise spline state representation (Bibby and Reid, 2010; Furgale et al., 2012), and a Gaussian process in GPGN (Tong et al., 2013b).

An alternative approach to laser-based VO is also presented in Anderson and Barfoot (2013b), which employs a piecewise spline representation for relative continuous-time SLAM. This work utilizes high-framerate lidar data, and provides an interesting application of the continuous-time state estimation formulation by performing velocity estimation.

In summary, we address the issue of motion distortion by employing a significantly different state estimation method than previous applications in the literature. The primary contributions of this chapter are the application of GPGN to a real-world 3D estimation problem, and the experimental validation of our laser-based VO framework using a panning laser rangefinder.



(a) A vertically scanning panning laser rangefinder composed of a laser mounted on a panning unit.

(b) An overhead illustration depicting the side-to-side sweep pattern used to obtain VO data. The vertically scanning laser rangefinder alternates between clockwise and counterclockwise sweeps while the rover is in motion.

Figure 6.2: The intended hardware and sweep configuration for 3D laser-based VO.

6.3 Architecture Overview

The primary sensor configuration we consider is a vertically scanning panning laser rangefinder, as depicted in Figure 6.2(a). Since we require rapid localization updates for control, we configure the payload to perform short, continuous, side-to-side sweeps while the rover is in motion. This sweep pattern is illustrated in Figure 6.2(b).

The processing pipeline, depicted in Figure 6.3, resembles that of standard appearance-based VO. First, we construct intensity images. Next, sparse visual features are detected and matched between consecutive frames, and outlier rejection is performed for robustness. The feature tracks are then used for state estimation, resulting in a 3D motion estimate. The following sections describe each of these steps in further detail.

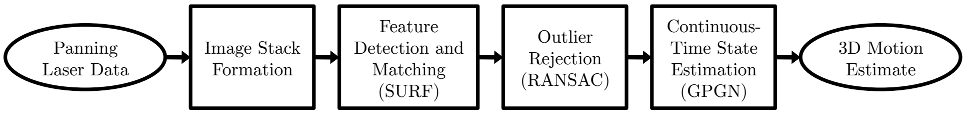

Panning Laser Data
Image Stack Formation
Feature Detection and Matching (SURF)
Outlier Rejection (RANSAC)
Continuous-Time State Estimation (GPGN)
3D Motion Estimate

Figure 6.3: The laser-based VO pipeline utilized in this chapter. First, intensity images are constructed from the laser data. Next, features are detected and matched between consective frames, and outlier rejection is performed for robustness. These feature tracks are then fed as a batch to the estimator, producing a motion estimate.

6.4 Feature Measurements

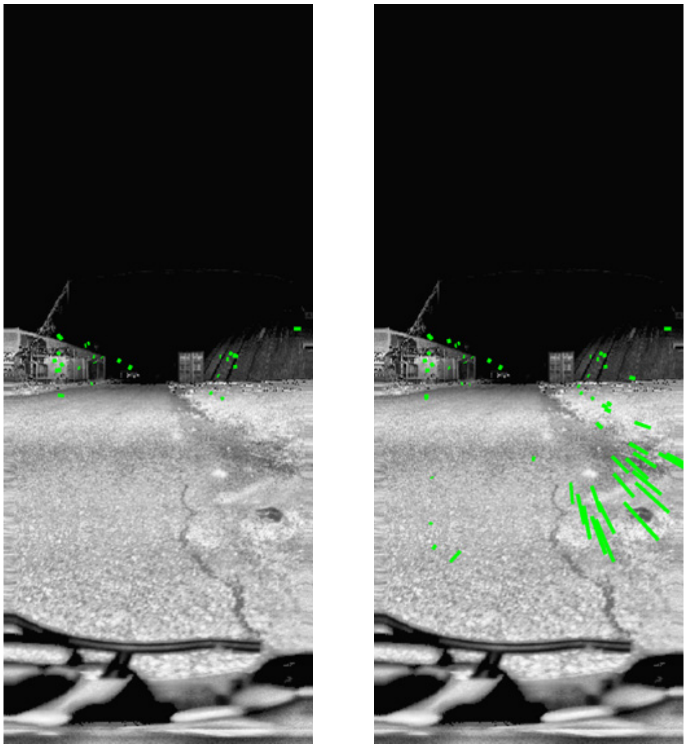
The measurement generation process is based upon previous work conducted by McManus et al. (2011), and refinements by Anderson et al. (2013). While some additional modifications were required for image formation, the overall framework remains the same. As a result, we only provide a brief overview in this section.

We begin with the formation of intensity images. These are constructed by simply placing the raw laser data into a 2D array, and scaling to produce 8-bit images. We obtain distinguishing contrast by applying contrast-limited adaptive histogram equalization (CLAHE) (Zuiderveld, 1994), but average over multiple tile configurations to avoid introducing artifacts. In addition, a smoothing step is performed using a Gaussian low-pass filter to reduce spatial aliasing effects (McManus et al., 2011). Since these images are used by the rest of the framework, improper image correction can lead to catastrophic algorithmic failures. The challenges involving image formation are discussed in further detail in Section 6.7.1.

As can be seen in Figure 6.1(a), the result of this process resembles a greyscale camera image. Therefore, we are able to utilize sparse appearance-based feature detectors common in the computer vision literature. In our implementation, we use a GPU-accelerated implementation of SURF (Bay et al., 2008). Some sample SURF features are depicted in Figure 6.1(b). Though the entire image is subject to motion distortion, each interest point is detected locally. As a result, the motion effects are small for each feature.

In addition to the intensity image, azimuth, elevation, range, and time images are also formed, creating an *image stack* (McManus et al., 2011). Since features are detected at subpixel locations, the image stack allows for subpixel lookup of these quantities through bilinear interpolation.

As with camera sensors, the measurements produced by laser rangefinders need to be calibrated. This was addressed by gathering a series of checkerboard images to perform characterization of the geometric distortions (Dong et al., 2013). Correcting for these distortions results in data that fit an idealized spherical camera model. Since the alternating sweep scan pattern may introduce additional distortion effects, the feature descriptors are matched between consecutive frames, as well as between alternating frames. Outlier matches are then validated using the Random Sample and Consensus (RANSAC) (Fischler and Bolles, 1981) algorithm. The resulting measurements and matches are the inputs for estimation.



(a) Feature tracks obtained from the standard rigid formulation.

(b) Feature tracks obtained from a motion-compensated formulation.

Figure 6.4: The inlier feature tracks obtained after applying RANSAC on candidate matches between consecutive frames. Due to motion distortion, the standard rigid transformation employed in RANSAC incorrectly rejects many valid matches. This issue was overcome by subdividing the image and introducing a piecewise constant velocity model into RANSAC (Anderson and Barfoot, 2013a)

.

However, we found that many valid matches were incorrectly rejected by a standard RANSAC implementation. This can be attributed to the fact RANSAC employs a rigid transformation to classify candidate matches. As a result, the uneven motion distortion present in the imagery and measurements resulted in many false positives. This was overcome by subdividing the image and introducing a simple piecewise constant velocity model into the RANSAC transformation (Anderson and Barfoot, 2013a). The improvement in classification performance is shown in Figure 6.4, where we show the inlier matches produced by both approaches.

6.5 Continuous-Time State Estimation

In this section, we address how to obtain motion estimates using the feature measurements. Following the theme of this thesis, we approach this task as a batch estimation problem, where a set of measurements are obtained over a period of time, and we seek to estimate the continuous rover trajectory during the time window. A sliding window approach is employed to limit the computational requirements, while still providing a smooth estimate.

To perform continuous-time state estimation, we utilize the GPGN algorithm from Chapter 3, and the 3D state parametrization described in Chapter 4. Once again, we restate some key equations in this section for convenience.

We represent the rover pose by the time-varying state component, and the feature locations as the time-invariant state component. In addition to the sliding window formulation, we also identify key poses for estimation, and employ GP interpolation to obtain intermediate poses at the feature measurement times. This approach ensures computational tractability. In our implementation, the key poses are defined to be at the midpoint acquisition time of each intensity image.

In general, the use of additional key poses provides greater estimate accuracy. However, a practical limit may be reached when the number of states employed sufficiently represents the underlying rover trajectory. While future developments may require a detailed study of the performance tradeoffs, our parametrization of one pose per image was chosen for simplicity in implementation.

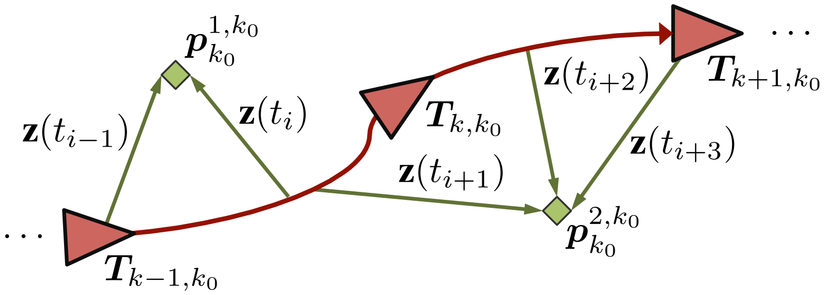


Figure 6.5: An illustration of the multiframe state estimation formulation. The state consists of an underlying time-varying state anchored by rover poses at key instants of time, $\boldsymbol{T}_{k,k_0}$, and the feature positions, $\boldsymbol{p}_{k_0}^{j,k_0}$. GP interpolation is utilized to obtain intermediate poses at the measurement times, and a sliding window approach is employed to maintain computational tractability.

For clarity, we provide an illustration summarizing our problem formulation in Figure 6.5. This approach differs from prior interpolation methods (Dong and Barfoot, 2012; Hedborg et al., 2012) because multiple poses are considered in GP interpolation, instead of simply the adjacent ones. This allows for a richer state representation with fewer key poses.

6.5.1 System Models

We begin by defining our state parametrizations. Though we are just interested in the rover trajectory from VO, navigation in unknown terrain requires us to simultaneously construct a map to determine the relative motion between frames. As a result, we treat the multiframe state estimation task as a batch SLAM problem, and estimate both the rover poses and the feature positions.

As described in Chapter 4, we express these quantities of interest with respect to an estimate reference frame, $\underrightarrow{\mathcal{F}}_{k_0}$, parametrize the rover pose at time t_k with a transformation matrix, $\boldsymbol{T}_{k,k_0}$, and represent the position of feature j with a homogeneous coordinate vector, $\boldsymbol{p}_{k_0}^{j,k_0}$. For the Gauss-Newton algorithm, these states are perturbed using

$$\boldsymbol{T}_{k,k_0} = e^{-\delta\boldsymbol{\pi}_{k,k_0}^{\boxplus}} \overline{\boldsymbol{T}}_{k,k_0}, \tag{6.1a}$$

$$\boldsymbol{p}_{k_0}^{j,k_0} = \overline{\boldsymbol{p}}_{k_0}^{j,k_0} + \mathbf{D}\,\delta\boldsymbol{\epsilon}_{k_0}^{j,k_0}, \tag{6.1b}$$

where $\bar{(\cdot)}$ denotes the current state estimate, and $\delta\boldsymbol{\pi}_{k,k_0}$ and $\delta\boldsymbol{\epsilon}_{k_0}^{j,k_0}$ are the rover pose and feature position perturbations, respectively. These expressions are used to obtain linearized measurement error models, and to compute the optimal updates that progressively bring us closer to the minimal estimate.

Next, we define the GP prior on the time-varying state component. Since a GP describes a distribution over a vector space, we utilize a vector parametrization of the state. That is, we define $\boldsymbol{\pi}(t_k)$ to be the vector representation corresponding to $\boldsymbol{T}_{k,k_0}$, and we utilize a CWNA model (Bar-Shalom et al., 2002) for the time-varying state, where

$$\ddot{\boldsymbol{\pi}}(t) \sim \mathcal{GP}\left(\mathbf{0}, \mathbf{W}\,\delta(t-t')\right), \tag{6.2}$$

$\mathbf{W}$ is the power spectral density matrix, and $\delta(\cdot)$ is the Dirac delta function. As discussed in Chapter 3, a more appropriate model may be used, but this simple model provides ease of implementation. Integrating the mean and covariance functions twice produces the GP for $\boldsymbol{\pi}(t)$, where

$$\boldsymbol{\pi}(t) \sim \mathcal{GP}(\boldsymbol{\mu}(t), \boldsymbol{\mathcal{K}}(t,t')), \tag{6.3a}$$

$$\boldsymbol{\mu}(t) = \boldsymbol{\pi}(0), \tag{6.3b}$$

$$\boldsymbol{\mathcal{K}}(t,t') = \mathbf{W}\left(\frac{\min(t,t')^2 \max(t,t')}{2} - \frac{\min(t,t')^3}{6}\right). \tag{6.3c}$$

The derivation of this covariance function is provided in Appendix A. Though this covariance function is fully dense, the number of features greatly outnumbers the number of estimated poses in our scenario. As a result, the overall system remains sparse.

While we have defined the GP prior using a vector parametrization for clarity, in implementation, we perform computations using a combination of transformation matrices and vector representations. This is based upon the approach presented in Chapter 4, where transformation matrices are used for expressions involving the state, and a vector parametrization is used for expressions involving small perturbations. For example, the mean pose is stored as a transformation matrix, and the logarithmic map is employed when a vector representation of the difference between two transformations is required.

The vector difference is utilized in the GPGN update (3.35) and prediction (3.49) equations to account for the deviation between the current estimate and the mean function. It is important to note that a discontinuity occurs on either end of the angular range, which violates the vector space assumptions employed by GPGN. While large rotation changes between consecutive frames do not tend to occur in the case of VO for a mobile robot, these discontinuities can also be avoided by defining a mean function that better captures the rover trajectory.

For the feature measurements, we utilize a generic camera model to define the observation of feature j with respect to pose at time t_k as

$$\mathbf{z}_{jk} := \mathbf{f}\left(\boldsymbol{T}_{k,k_0}\boldsymbol{p}_{k_0}^{j,k_0}\right) + \mathbf{n}_{jk}, \quad \mathbf{n}_{jk} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{jk}), \tag{6.4}$$

where the noise parameter, $\mathbf{n}_{jk}$, is additive, zero-mean, and Gaussian-distributed, with a covariance of $\mathbf{R}_{jk}$. Since our panning laser rangefinder provides azimuth, elevation, and range data, we transform the homogeneous point to Cartesian coordinates and utilize a spherical camera model for $\mathbf{f}(\cdot)$. That is,

$$\mathbf{f}(\boldsymbol{p}) := \begin{bmatrix} \operatorname{atan2}(p_2, p_1) \\ \operatorname{atan2}(p_3, \sqrt{p_1^2 + p_2^2}) \\ \frac{\sqrt{p_1^2+p_2^2+p_3^2}}{p_4} \end{bmatrix}. \tag{6.5}$$

We obtain a linearized measurement error model by applying the linearized camera model expression from Chapter 4 to result in

$$\mathbf{e}_{jk} \approx \mathbf{z}_{jk} - \mathbf{f}(\overline{\boldsymbol{p}}_k^{j,k}) - \mathbf{F}_{jk} \begin{bmatrix} \overline{\boldsymbol{p}}_k^{j,k^{\boxminus}} & \overline{\boldsymbol{T}}_{k,k_0}\mathbf{D} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{\pi}_{k,k_0} \\ \delta\boldsymbol{\epsilon}_{k_0}^{j,k_0} \end{bmatrix}, \quad \mathbf{F}_{jk} := \left.\frac{\partial \mathbf{f}}{\partial \boldsymbol{p}}\right|_{\overline{\boldsymbol{p}}_k^{j,k}}, \tag{6.6}$$

where the camera model Jacobian, $\mathbf{F}_{jk}$, is

$$\mathbf{F}_{jk} = \begin{bmatrix} -\frac{p_2}{p_1^2+p_2^2} & \frac{p_1}{p_1^2+p_2^2} & 0 & 0 \\ -\frac{p_1 p_3}{\left(p_1^2+p_2^2+p_3^2\right)\sqrt{p_1^2+p_2^2}} & -\frac{p_2 p_3}{\left(p_1^2+p_2^2+p_3^2\right)\sqrt{p_1^2+p_2^2}} & \frac{\sqrt{p_1^2+p_2^2}}{p_1^2+p_2^2+p_3^2} & 0 \\ \frac{p_1}{p_4\sqrt{p_1^2+p_2^2+p_3^2}} & \frac{p_2}{p_4\sqrt{p_1^2+p_2^2+p_3^2}} & \frac{p_3}{p_4\sqrt{p_1^2+p_2^2+p_3^2}} & -\frac{\sqrt{p_1^2+p_2^2+p_3^2}}{p_4^2} \end{bmatrix}. \tag{6.7}$$

6.5.2 Optimization

Given definitions for the GP prior and measurement model, we seek the optimal estimate by following the Gauss-Newton procedure detailed in Chapter 3. We identify the key poses for estimation, and employ GP interpolation to obtain the intermediate poses. For each Gauss-Newton iteration, the linear system of equations is constructed using (6.6), and evaluations of the covariance function (6.3c) at the measurement times.

As stated earlier, the covariance function is fully dense. However, the large number of features still results in a sparse system. This is visually confirmed by the sparsity pattern depicted in Figure 6.6. As a result, we can solve the system efficiently using the Schur complement to obtain the optimal perturbations. These perturbations are then used to update the state estimates using the constraint-sensitive update equations (6.1), and the process repeats until convergence.

6.5.3 Sliding Window

Since computational limits prevent us from estimating the entire rover trajectory as a single problem, we utilize the batch formulation of GPGN in a multiframe sliding window implementation. In this approach, only a small portion of the trajectory is considered in each estimation problem. This consists of a number of fixed poses to maintain consistency, as well as some free poses to be estimated. After convergence, the feature position estimates are discarded, and the estimation window is slid forward. Alternative formulations such as the Sliding Window Filter (Sibley et al., 2010a) maintain a local map by conditioning out old poses and landmarks. While this allows the estimator to benefit from the improved map estimates, the sparsity in the linear system is lost.

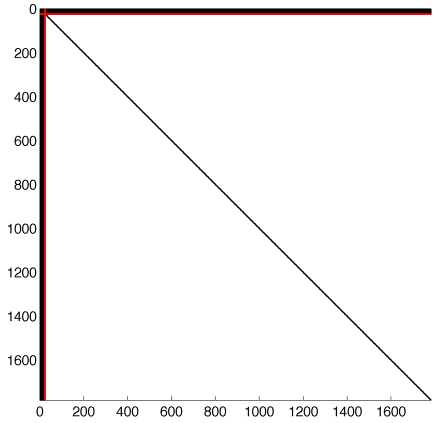


Figure 6.6: The sparsity pattern of a typical inverse covariance matrix (3.36) constructed during our laser-based VO experiments. The red lines indicate the partitioning between rover poses and feature positions. While the covariance function is fully dense, the feature positions dominate the estimation problem. As a result, the system remains sparse.

The primary consideration in implementing a sliding window is the tradeoff between accuracy and computation time. Though the computational requirements of batch estimation scale with the number of measurements, maintaining a fixed window size allows for approximately constant computation. However, since GP interpolation utilizes multiple frames, a sufficiently large number of poses is required to maintain the richness of the state representation. In our implementation, we used a sliding window configuration of 5 fixed frames and 3 free poses. This configuration is illustrated in Figure 6.7.

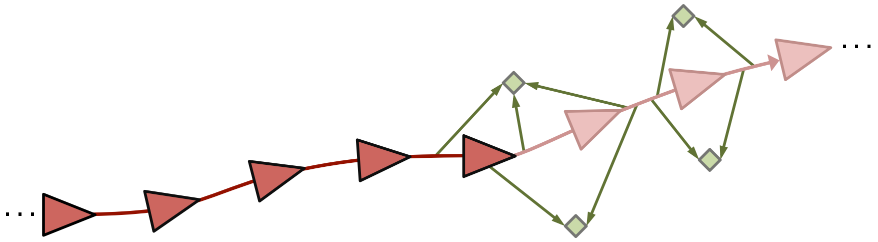


Figure 6.7: An illustration of the sliding window configuration, where rover poses are represented by red triangles, and feature positions by green diamonds. Each window consists of a number of fixed poses, depicted by the dark-coloured objects, and a number of free poses, depicted by the light-coloured objects. The fixed poses are obtained from the previous window, and incorporated into the estimation problem to maintain a smooth trajectory. The estimated state is composed of the free poses and feature positions. After convergence, the window is slid forward one pose, and the process is repeated.

6.6 Experimental Validation

The experimental validation of the laser-based VO framework is split into two parts. The first dataset consists of a 365m traverse collected in an outdoor environment, which allows us to evaluate the performance over a long distance. This is supplemented by the second dataset, which is composed of a series of short traverses in planetary analogue terrain to reflect the intended mapping scenario. Additional experimental validation of the framework involving a different high-framerate scanning lidar operating for 1.1km in natural, unstructured terrain can be found in Tong and Barfoot (2013b).

For comparison, we implemented our proposed GPGN formulation alongside the conventional discrete-time batch Gauss-Newton approach. In this approach, the timestamps of the feature observations are replaced by the assumption that features detected in a single intensity image all arrive at the same time. In other words, the motion distortion effects are ignored, and the rover is assumed to be stationary during a single image acquisition. While this does not reflect the actual acquisition process, this compensation-free approach provides a point of reference that reflects the current discrete-time state estimation literature. Future work will involve a comparison between GPGN and the pose interpolation methods employed by Hedborg et al. (2012) and Dong and Barfoot (2012). In both cases, a Huber M-estimator (Huber, 1981) was incorporated for additional robustness against measurement outliers, and a maximum motion threshold was enforced to reject unrealistic pose estimates suggested by poor feature matches.

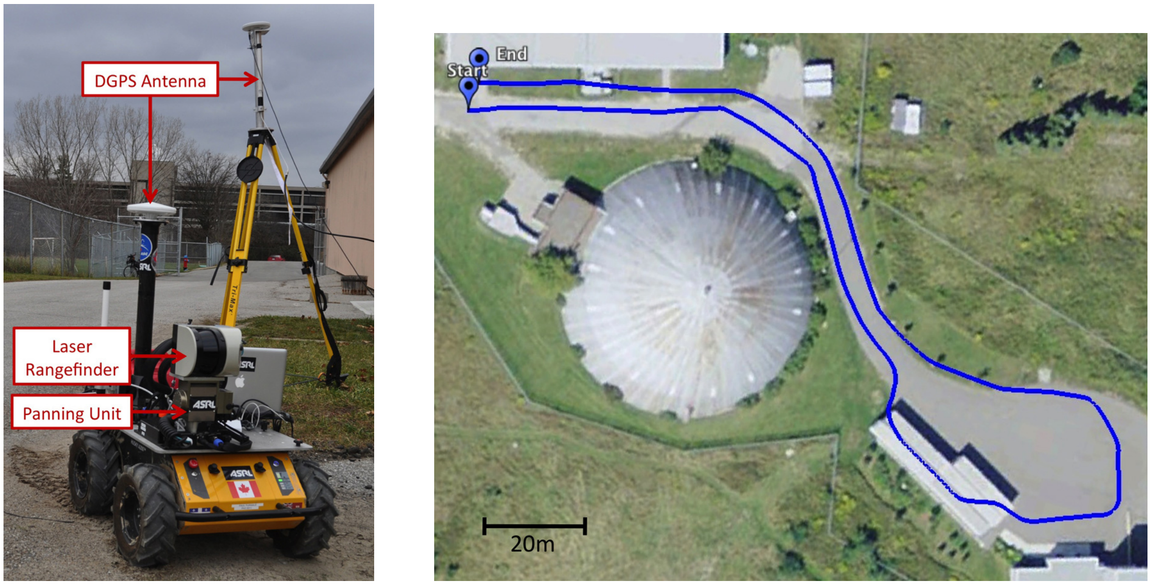
Tuning was conducted using a small section of a representative traverse, where the best estimator performance was achieved by a sliding window configuration of 5 fixed and 3 free poses. Similarly, the sensor noise covariance, M-estimator threshold, and GP hyperparameters were determined using a combination of the manufacturer's specifications, ground truth data, and manual experimentation. Though our GP prior does not model the nonholonomic restrictions of the rover platform, we implicitly limited the range of motion estimates by penalizing large pitch, roll, and lateral motions through appropriate weighting of the GP hyperparameters. The sliding window and sensor noise parameters were kept identical for both algorithms and both test sites.

6.6.1 Outdoor Traverse

The first dataset was obtained outdoors at the University of Toronto Institute for Aerospace Studies (UTIAS). This test site offered a large area for conducting a lengthy traverse, and the ability to obtain continuous ground truth. As a result, it provided a good opportunity to evaluate the performance and error growth of the estimation algorithms.

The rover used to gather the dataset, depicted in Figure 6.8(a), was equipped with a laser rangefinder mounted on a panning unit. The vertically scanning laser had an angular range of $[-95°, 95°]$, and was configured to provide scan lines at 50Hz with a $\frac{1}{3}^\circ$ point spacing. The panning unit was configured to pan between $[-46°, 46°]$, at an approximate speed of 20°/s. This resulted in a $92\times190°$ field of view, and an acquisition time of about 4.6s for each 220×571 pixel image. Ground truth data was provided by a combination of a DGPS base station and an onboard antenna.

The dataset was collected in the early evening over the course of 2h. During data collection, the rover was manually driven at a speed of 5cm/s over paved roads in a looping traverse, as depicted in Figure 6.8(b). While slow, the combination of this speed and the slow scan rate still produced challenging motion distortions. With an acquisition time of 4.6s, the 5cm/s speed translated into 23cm of motion within each image. In total, the traverse produced 1628 image stacks over 365m of travel.



(a) The Clearpath Husky A200 rover used to gather the dataset. The relevant payloads are the laser rangefinder mounted on a panning unit, and the DGPS antennas for ground truth data.

(b) The path taken by the rover as calculated by post-processed DGPS. The rover was driven manually in the early evening over paved roads bordered by dirt and light vegetation, resulting in a total of 365m of travel. The stack of containers intersecting the path were not present during the experiment.

Figure 6.8: The hardware and test site for the outdoor traverse.

The resulting laser-based VO estimates are depicted in Figure 6.9. Since the VO estimates and ground truth data did not share a common reference frame, the first 60m of each estimate was used for alignment. As can be seen, the continuous-time GPGN formulation greatly outperformed the discrete-time compensation-free approach. However, it should be noted that GPGN also consistently underestimated the turns. This can be attributed to the smoothing behaviour of the GP prior, as well as the difficulty in feature tracking during yaw motions. These issues are discussed in Section 6.7.1.

We quantify the growth in estimation error by plotting the translation errors against the actual distance travelled in Figure 6.10. These results confirm our qualitative observations that GPGN is significantly more accurate. However, some beneficial effects in error reduction were experienced due to the overlapping path.

Finally, we provide timing analysis to compare the two estimator implementations[1]. In total, the compensation-free approach took 62mins for state estimation, and GPGN took 76mins. Dividing these durations by the number of frames resulted in an average of 2.3s per estimation window in the compensation-free approach, and 2.8s for GPGN.

[1] Timing information recorded on a MacBook Pro with a 2.66GHz Core 2 Duo and 4GB of 1067MHz DDR3 RAM in Matlab, utilizing both cores.

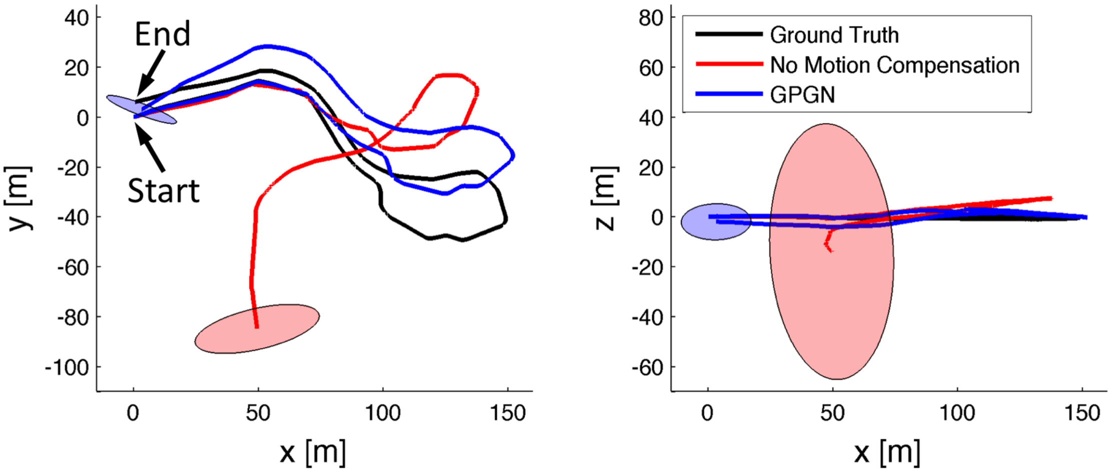


Figure 6.9: An overhead and side plot of the VO estimates over the 365m traverse. The ground truth GPS data is depicted in black, the compensation-free estimate in red, and our GPGN estimate in blue. Projected 3σ covariance envelopes are represented by shaded ellipses at the end of each traverse to illustrate the estimator consistency. The first 60m of each estimate was used for alignment, and as can be seen, the continuous-time GPGN formulation greatly outperforms the discrete-time compensation-free approach.

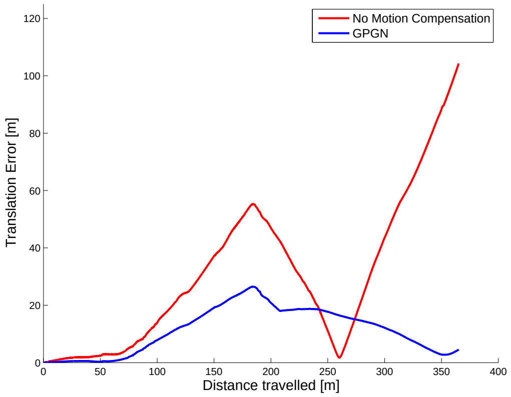


Figure 6.10: A plot of the translation estimate errors compared to the actual distance travelled, where the first 60m of each trajectory was used for alignment to the ground truth data. As can be seen, GPGN greatly outperforms the compensation-free approach.

While this timing analysis does not include feature extraction and matching, the processing time is much less than the acquisition time of 4.6s per frame. It is expected that implementation in C++ will make this a viable choice for online operation. Furthermore, since the increase in computation time from the conventional discrete-time approach to the continuous-time GPGN formulation is small, it provides promise for addressing online operation by simply augmenting existing online discrete-time state estimation code.

6.6.2 UTIAS Indoor Rover Test Facility

The second dataset was obtained at the UTIAS indoor rover test facility. The facility, depicted in Figure 6.11, consists of gravel spread in a 40m diameter circular workspace, which emulated scaled planetary hills and ridges. This is the same test facility used to validate the mapping framework in Chapter 5. To reflect the intended worksite mapping scenario, we revisited this planetary analogue site to demonstrate that the laser-based VO framework is suitable for providing the pose-to-pose estimates between scan stops.

We used the same rover platform as depicted in Figure 6.8(a), but removed the DGPS antenna to eliminate self-occlusions. The scan configurations remained the same, and the rover was driven manually at 5cm/s in an exploratory traverse.

The dataset consists of a series of nine short traverses conducted between ten scan stops. At these stops, 360×190° scans were obtained, and the ground truth poses were computed by the identifying known landmarks outside the workspace (Tong and Barfoot, 2011b). In total, 671 image stacks were produced over approximately 130m of travel.

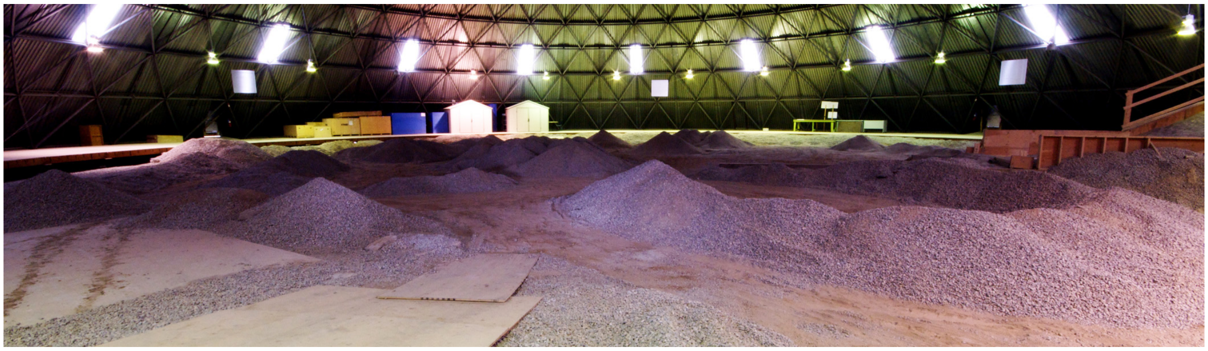


Figure 6.11: An image of the UTIAS indoor rover test facility. The terrain consists of gravel spread in a 40m diameter circular workspace, emulating scaled planetary features.

The laser-based VO estimates are depicted in Figure 6.12, where we have overlaid the estimated trajectories onto a ground truth map constructed from the mapping scans. To illustrate endpoint accuracy, each traverse was initialized at the ground truth poses. Once again, GPGN outperforms the compensation-free approach. This is quantified by the average RMS errors, where the compensation-free approach achieved an average endpoint error of 1.1m in translation and 7.6° in rotation, compared to an average endpoint error of 0.6m in translation and 6.9° in rotation for GPGN.

The timing results for this dataset were similar to the outdoor traverse[2]. In total, the compensation-free approach took 42mins for state estimation, and GPGN took 62mins. This resulted in an average of 3.7s per estimation window in the compensation-free implementation, and 5.6s for GPGN. The increase in computation time per estimation window can be attributed to an increase in feature count as compared to the outdoor scenario.

Finally, we demonstrate the applicability of the VO framework by incorporating the pose-to-pose estimates produced by GPGN into the mapping framework presented in Chapter 5. The estimated map was constructed by overlaying the dense scans according to the pose estimates, and utilizing a metascan ICP (Chen and Medioni, 1992) approach for refinement. The results of this procedure can be seen in Figure 6.13. While improved performance can be attained through additional data collection, these results validate our combined navigation and mapping approach.

[2] Timing information recorded using the same computer as the outdoor dataset.

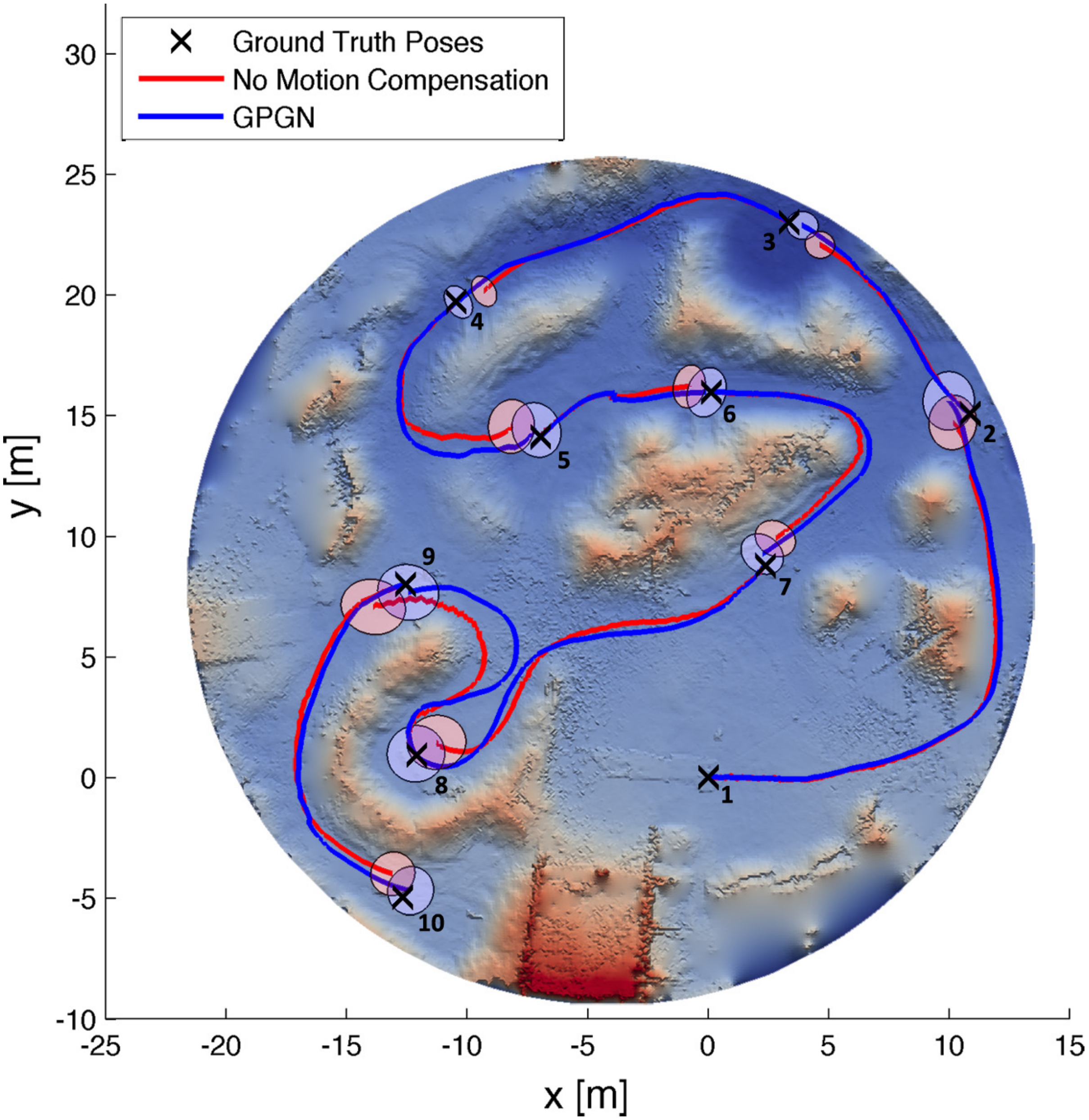
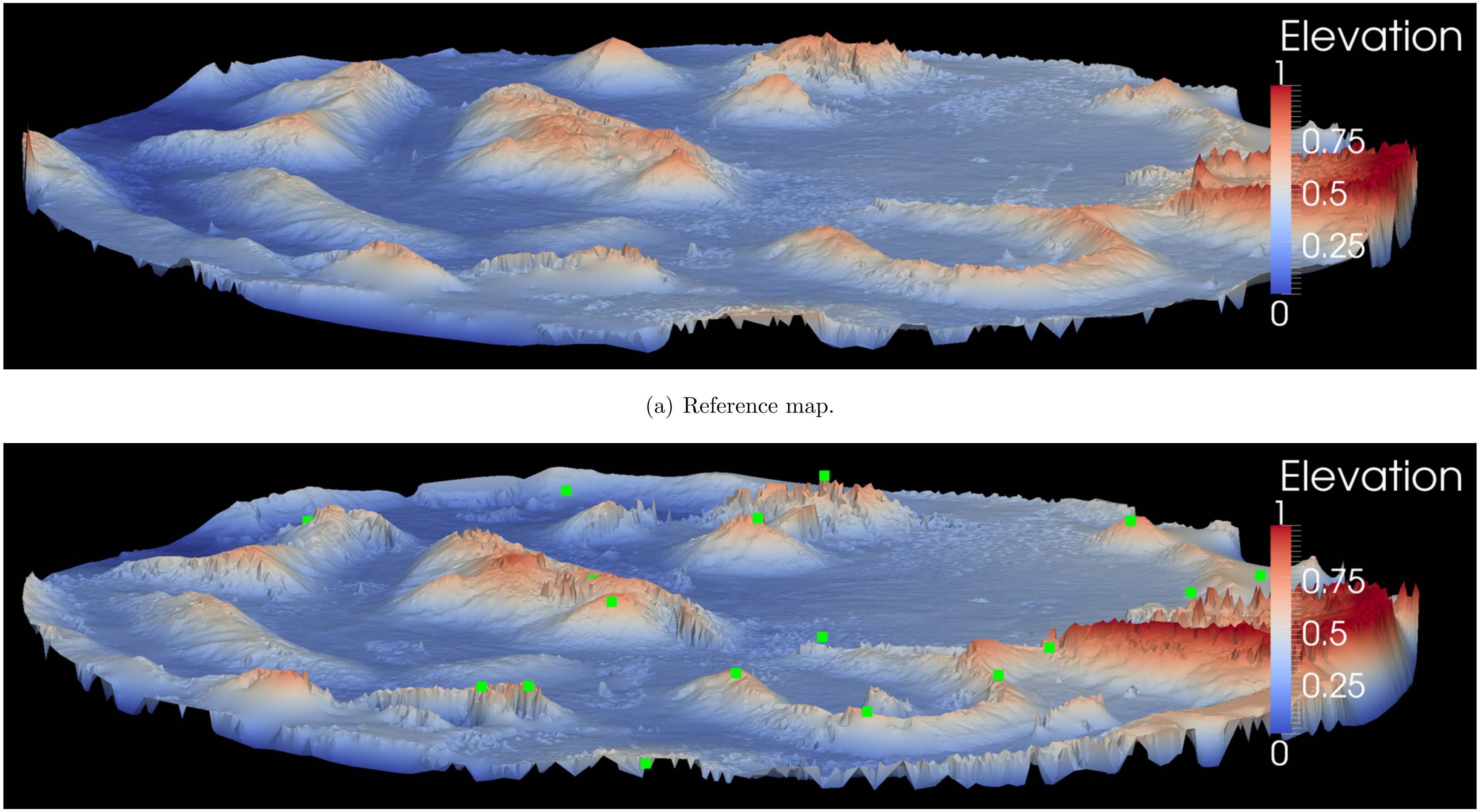


Figure 6.12: An overhead plot of the VO estimates overlaid on a ground truth map, where we illustrate endpoint accuracy by initializing each traverse at the ground truth poses. The shaded ellipses represent the projected 3σ covariance envelopes.



(a) Reference map.

(b) Estimated map constructed using the mapping framework and GPGN VO.

Figure 6.13: Meshes created from the reference and estimated point clouds, where interpolation was employed to fill in the occluded peaks. In (b), the green squares mark the estimated sparse feature positions. As can be seen, the two maps are visually similar, validating our combined navigation and mapping approach.

6.7 Discussion

Although the overall performance of the laser-based VO framework was positive, some shortcomings warrant further analysis. In particular, we address the difficulty in feature tracking during turns, and challenges in image formation. This is presented through discussion of the failure modes, and possibilities for future improvements.

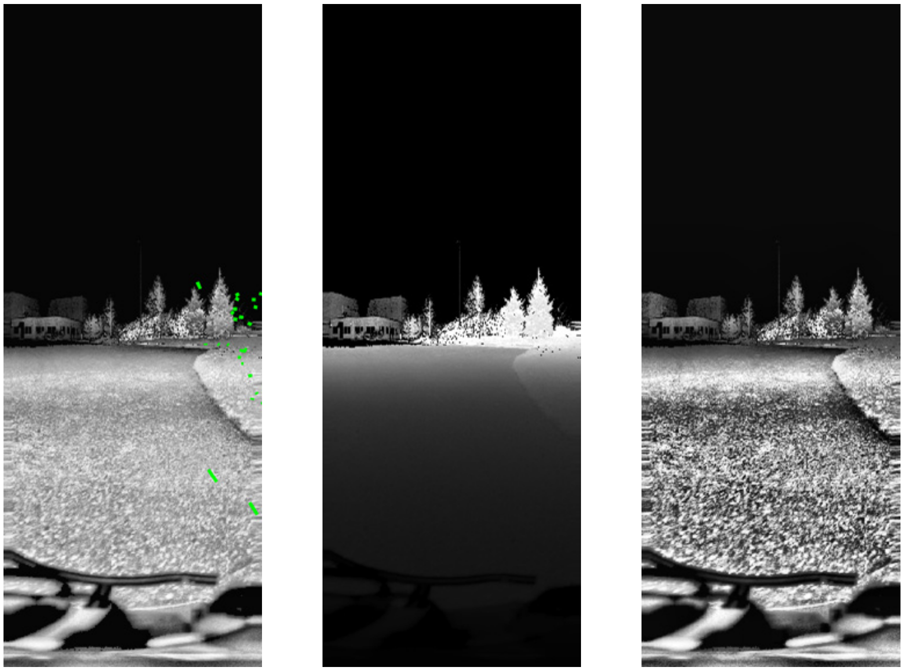
6.7.1 Failure Modes

One of the key issues in implementation was feature tracking in intensity imagery. While intensity images may resemble greyscale camera imagery, the alternating sweep pattern and significantly lower resolution presented many challenges.

For example, the right-hand-side of images constructed from a clockwise sweep followed by a counterclockwise sweep are obtained close together in time. As a result, they share similar appearances, and provide strong rigid matches. Though motion-compensation was introduced into RANSAC to account for distortion, the combined rotation from yaw motions and the sweep pattern occasionally produced situations where only a thin vertical strip of matches were identified between frames. This is depicted in Figure 6.14(a). In these scenarios, the estimator is free to pivot, producing large yaw estimates. This issue was overcome by obtaining additional matches between alternating frames, and restricting yaw motions through tuning of the GP hyperparameters. While this provided robust performance, it tended to conservatively underestimate turns.

Some improvements may be achieved through modifications in image formation. However, the tradeoff between highlighting distinct features and introducing artifacts can be difficult. Since intensity information is dependent on range, we attempted to correct for the power dissipation by multiplying the intensity returns by the range values, and rescaling into an 8-bit image. The result of this is shown in Figure 6.14(b), where we see that this procedure failed to produce an image with any useful texture. This failure may be attributed to an unknown mapping or the limited resolution of the intensity data.

Since range correction was ineffective, we performed contrast correction to enhance the intensity values relative to their local neighbourhoods. While successful feature tracking was attained using our implementation parameters, overcorrection was a dangerous possibility. In Figure 6.14(c), we see that overcorrection produced distinct texture, but also introduced artificial elements by enhancing noise. Since these images are used by the rest of the VO framework, a careful balance must be struck for robust operation.



(a) Poor feature track distribution.

(b) Indistinct linear range correction.

(c) Noisy contrast overcorrection.

Figure 6.14: A series of three intensity images constructed from the same laser data, which highlights some challenges in image formation and feature matching. In (a), the alternating sweep pattern led to a thin vertical strip of feature matches. In these scenarios, the estimator produced large yaw estimates. In (b), we attempted to correct for the range dependence of intensity data by performing a simple linear range correction. However, this did not produce an image with any useful texture. Finally, in (c), contrast overcorrection provided distinct texture, but also introduced artificial elements.

6.7.2 Possible Improvements

Since the failure modes identified were largely due to shortcomings in the feature extraction and mapping procedure, improvements at that stage should have a significant impact on the overall performance of the VO framework. The challenges introduced by the sweep pattern may be avoided by redesigning the laser hardware into a spinning configuration, which should produce consistent distortion within each image. In addition, future work should include revisiting the choice of SURF as the feature detector. SURF was originally designed to identify circular blobs, and as a result, may be inappropriate for distorted imagery. An alternative choice of image features such as affine invariant features (Mikolajczyk and Schmid, 2004) may produce better results.

Furthermore, we utilized a simple GP in our formulation that did not consider any physical properties of the rover platform. While we were able to implicitly model the nonholonomic restrictions by penalizing large pitch, roll, and lateral translations, more accurate estimates may result by utilizing a more complex kernel function that incorporates a parametric model into the mean function (Ko and Fox, 2009; Guizilini and Ramos, 2012).

6.8 Conclusion

In summary, we have presented a novel 3D laser-based VO algorithm in this chapter. We leverage existing successes from the computer vision literature by taking an appearance-based approach, where intensity images are constructed, and sparse visual features are tracked for motion estimation. To overcome the motion distortion effects, we utilize GPGN for continuous-time state estimation, which considers the timestamps of the feature measurements, and maintains an underlying continuous state.

The primary contribution of this chapter is a practical application of the GPGN algorithm introduced in Chapter 3. This state estimation algorithm was incorporated into a laser-based VO framework, and validated using a 365m traverse, as well as a planetary worksite mapping task. Through this analysis, some limitations in estimator performance were identified, but it was shown that that GPGN greatly outperforms the compensation-free approach in estimate accuracy. Furthermore, the planetary worksite map produced by the combination of laser-based VO estimates and peak features in the mapping framework validated our combined navigation and mapping approach.

Chapter 7

Conclusion

In conclusion, we have incorporated a panning laser rangefinder into working mapping and navigation frameworks suitable for operation in natural, unstructured, 3D terrain. The combination of these two algorithmic components enable planetary worksite operations. A number of novel contributions were made over the course of this thesis. In this chapter, we summarize these contributions, speculate about future work, and provide some concluding thoughts.

We began with a mathematical review of the conventional method for discrete-time batch state estimation in Chapter 2, where we derived the discrete-time batch Gauss-Newton algorithm. This was followed by the introduction of Gaussian Process Gauss-Newton (GPGN) in Chapter 3, a novel algorithm for nonparametric, continuous-time, nonlinear, batch state estimation. Two derivations were provided, as well as approximation methods to maintain computational efficiency. Furthermore, experimental validation was conducted by comparing the discrete-time batch Gauss-Newton approach to GPGN for a 2D SLAM problem. The publications associated to these chapters are:

- Tong, C., Furgale, P., and Barfoot, T. D. (2012b). Gaussian Process Gauss-Newton: Non-parametric state estimation. In *Proceedings of the 9th Conference on Computer and Robot Vision (CRV)*, pages 206–213, Toronto, Ontario, Canada.
- Tong, C., Furgale, P., and Barfoot, T. D. (2013b). Gaussian Process Gauss-Newton for non-parametric simultaneous localization and mapping. *International Journal of Robotics Research (IJRR)*, 32(5):507–525.

Since the batch state estimation algorithms were derived using linear algebra, complexities arise when attempting to apply the algorithms to the 3D domain. Chapter 4 addressed this issue by presenting a suitable 3D state parametrization in the form of transformation matrices, which provides a local vector space for estimation. This state representation was then used to perform linearization of the measurement models used in this thesis.

The mathematical preliminaries were then incorporated into a mapping framework in Chapter 5, where we autonomously created accurate maps of natural terrain using data obtained at two different planetary analogue test sites. We employed a stop-scan-go traverse to obtain long-range 360° scans at infrequent stops, and utilized a combination of sparse peak features, dense data, odometry, and heterogeneous outlier rejection in a discrete-time 3D batch SLAM formulation for alignment. The publications associated to this work can be divided into multiple categories. The first category is associated to the characterization of various heterogeneous measurement outlier rejection schemes initially performed to inform the future design decisions. The publications containing these characterization experiments are:

- Tong, C. and Barfoot, T. D. (2011a). Batch heterogeneous outlier rejection for feature-poor SLAM. In *Proceedings of the IEEE Conference on Robotics and Automation (ICRA)*, pages 2630–2637, Shanghai, China.
- Tong, C. and Barfoot, T. D. (2013a). Evaluation of heterogenous measurement outlier rejection schemes for robotic planetary worksite mapping. *Acta Astronautica*, 88:146–162.

The second category relates to the entire mapping framework, which was published as:

- Tong, C., Barfoot, T. D., and Dupuis, E. (2011). 3D SLAM for planetary worksite mapping. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Invited Symposium on "Field Robotics II"*, pages 631–638, San Francisco, CA, USA.
- Tong, C., Barfoot, T. D., and Dupuis, E. (2012a). Three-dimensional SLAM for mapping planetary work site environments. *Journal of Field Robotics, Special issue on "Space Robotics"*, 29(3):381–412.

Finally, the efforts towards experimental validation also led to a number of publications. These contributions include an infrastructure-based ground truth system using retroreflective landmarks, the application of the framework for a human-guided exploration task, and the experimental data. These were published as follows:

- Tong, C. and Barfoot, T. D. (2011b). A self-calibrating ground-truth localization system using retroreflective landmarks. In *Proceedings of the IEEE Conference on Robotics and Automation (ICRA)*, pages 3601–3606, Shanghai, China.
- Merali, R. S., Tong, C., Gammell, J., Bakambu, J., Dupuis, E., and Barfoot, T. D. (2012). 3D surface mapping using a semi-autonomous rover: A planetary analog field experiment. In *Proceedings of the 11th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, Turin, Italy.
- Tong, C., Gingras, D., Larose, K., Barfoot, T. D., and Dupuis, E. (2013c). The Canadian planetary emulation terrain 3D mapping dataset. *International Journal of Robotics Research (IJRR)*, 32(4):389–395.

To complement the mapping framework, we also addressed localization for navigation between scan stops in Chapter 6. We configured the panning laser to perform short, continuous sweeps while the rover was in motion, and employed a sparse appearance-based approach for motion estimation, where laser intensity images were constructed and utilized to compute VO estimates. The motion distortion issues were addressed by incorporating the timestamps of the feature measurements into a continuous-time GPGN formulation of the estimation problem. This resulted in a successful demonstration of laser-based VO in both an outdoor environment and in planetary analogue terrain. The publication associated to the laser-based VO algorithm is:

- Tong, C. and Barfoot, T. D. (2013b). Gaussian Process Gauss-Newton for 3D laser-based visual odometry. In *Proceedings of the IEEE Conference on Robotics and Automation (ICRA)*, pages 5184–5191, Karlsruhe, Germany.

A journal publication (Tong et al., 2013a) containing more extensive experiments, comparisons between pose interpolation methods, and a different high-framerate scanning lidar is forthcoming.

There are many possibilities for future work stemming from the material presented in this thesis. While these possibilities include the incremental improvements suggested in each chapter, there also major areas for further development and theoretical contribution.

In this thesis, we focused on the state estimation necessary to enable planetary worksite mapping and navigation. While we validated the underlying theory through experimentation, shortcomings in both the mapping and navigation frameworks were identified in the measurement generation process. In the case of mapping, feature scarcity resulted in unused scans, and poor data associations in navigation led to reduced accuracy during turns. Further investigation is necessary to determine the most suitable features for these scenarios, which may require new development tailored to laser data. For example, a feature may be detected and described using a combination of range and intensity data. Improvements to feature detection and matching will produce more accurate maps, and enable longer traverses.

GPGN presents a novel approach to state estimation, which considers the estimation problem in continuous time. While its derivation may appear to be complex, it is easily implemented by augmenting existing discrete-time state estimation algorithms by incorporating the additional regularization terms. Laser-based VO is only one novel application enabled by GPGN. Other possible applications include estimation using rolling shutter cameras, incorporating high-rate IMU measurements, and operation in fully 3D domains with aerial vehicles. Theoretical contributions may also be made in determining other physically motivated kernel functions that better capture the vehicle motion, and investigating adaptive pose placement that balances computational efficiency with representational fidelity.

Finally, although we considered the mapping and navigation tasks separately in this thesis, further integration between the two components may be conducted. For example, the long-range scans obtained during mapping could be used for localization during navigation. This would limit the estimator drift, but would require the ability to recognize distinctive elements of the scene from multiple viewpoints. Similarly, the mapping performance should benefit from the dense local data obtained during navigation. Tighter integration between these two components will enable autonomous exploration.

These possibilities for future work will lead to greater capability for autonomous mobile robots. I hope that continued research in this area will eventually make terrestrial and planetary autonomy a reality.

Appendix A

Physically Motivated Gaussian Processes

A.1 Gaussian Process for a Moving Point

In GPGN, the mean and covariance functions define the properties of the underlying process model that we wish to estimate. As a result, appropriate GP selection can greatly affect the performance of the algorithm. This appendix details the derivation of the physically motivated GP utilized in this thesis.

For clarity, we define

$$\mathbf{a}(t) := \ddot{\mathbf{x}}(t), \tag{A.1}$$

$$\mathbf{v}(t) := \dot{\mathbf{x}}(t), \tag{A.2}$$

$$\mathbf{d}(t) := \mathbf{x}(t), \tag{A.3}$$

and for simplicity, we will assume that the integration and initial conditions begin at $t = 0$.

A.1.1 General Derivation

We begin by showing how to obtain the stochastic process properties of the velocity and position for a moving point, starting with a GP model on the acceleration. This is followed by a specific example of the white noise prior in the following section.

A.1.1.1 Acceleration

We define the GP prior on acceleration as

$$\mathbf{a}(t) \sim \mathcal{GP}\left(\boldsymbol{\mu}_a(t), \boldsymbol{\mathcal{K}}_{aa}(t,t')\right), \tag{A.4}$$

$$\boldsymbol{\mu}_a(t) := E\left[\mathbf{a}(t)\right], \tag{A.5}$$

$$\boldsymbol{\mathcal{K}}_{aa}(t,t') := E\left[\left(\mathbf{a}(t) - \boldsymbol{\mu}_a(t)\right)\left(\mathbf{a}(t') - \boldsymbol{\mu}_a(t')\right)^T\right]. \tag{A.6}$$

A.1.1.2 Velocity

The relation between velocity and acceleration is

$$\mathbf{v}(t) = \int_0^t \mathbf{a}(\tau)d\tau + \mathbf{v}(0), \tag{A.7}$$

where $\mathbf{v}(0)$ is provided as an initial condition for the velocity. Using this expression, the mean function is

$$\begin{aligned}\boldsymbol{\mu}_v(t) &= E\left[\int_0^t \mathbf{a}(\tau)d\tau + \mathbf{v}(0)\right] &\text{(A.8)}\\ &= \int_0^t \boldsymbol{\mu}_a(\tau)d\tau + \boldsymbol{\mu}_v(0), &\text{(A.9)}\end{aligned}$$

and the covariance function is

$$\begin{aligned}\boldsymbol{\mathcal{K}}_{vv}(t,t') = E&\left[\left(\mathbf{v}(t) - \boldsymbol{\mu}_v(t)\right)\left(\mathbf{v}(t') - \boldsymbol{\mu}_v(t')\right)^T\right]\\ = &\int_0^{t'}\int_0^t E\left[\left(\mathbf{a}(\tau) - \boldsymbol{\mu}_a(\tau)\right)\left(\mathbf{a}(\tau') - \boldsymbol{\mu}_a(\tau')\right)^T\right] d\tau\, d\tau'\\ &+ \int_0^t E\left[\left(\mathbf{a}(\tau) - \boldsymbol{\mu}_a(\tau)\right)\left(\mathbf{v}(0) - \boldsymbol{\mu}_v(0)\right)^T\right] d\tau\\ &+ \int_0^{t'} E\left[\left(\mathbf{v}(0) - \boldsymbol{\mu}_v(0)\right)\left(\mathbf{a}(\tau') - \boldsymbol{\mu}_a(\tau')\right)^T\right] d\tau'\\ &+ E\left[\left(\mathbf{v}(0) - \boldsymbol{\mu}_v(0)\right)\left(\mathbf{v}(0) - \boldsymbol{\mu}_v(0)\right)^T\right]. \end{aligned} \tag{A.10}$$

We simplify this expression by assuming the acceleration and initial velocity terms are independent, resulting in

$$\boldsymbol{\mathcal{K}}_{vv}(t,t') = \int_0^{t'}\int_0^t \boldsymbol{\mathcal{K}}_{aa}(\tau,\tau')d\tau d\tau' + \boldsymbol{\mathcal{K}}_{vv}(0,0), \tag{A.11}$$

where $\boldsymbol{\mathcal{K}}_{vv}(0,0)$ is typically provided as an initial uncertainty for the estimation problem.

A.1.1.3 Position

Similarly, the relation between position and acceleration is

$$\begin{aligned}\mathbf{d}(t) &= \int_0^t \mathbf{v}(\tau)d\tau + \mathbf{d}(0)\\ &= \int_0^t\int_0^\tau \mathbf{a}(s)ds\,d\tau + \int_0^t \mathbf{v}(0)d\tau + \mathbf{d}(0).\end{aligned} \tag{A.12}$$

Using this relation, the mean function is

$$\begin{aligned}\boldsymbol{\mu}_d(t) &= E\left[\int_0^t\int_0^\tau \mathbf{a}(s)ds\,d\tau + \int_0^t \mathbf{v}(0)d\tau + \mathbf{d}(0)\right]\\ &= \int_0^t\int_0^\tau \boldsymbol{\mu}_a(s)ds\,d\tau + \boldsymbol{\mu}_v(0)t + \boldsymbol{\mu}_d(0),\end{aligned} \tag{A.13}$$

and the covariance function is

$$\begin{aligned}\boldsymbol{\mathcal{K}}_{dd}(t,t') &= E\left[\left(\mathbf{d}(t)-\boldsymbol{\mu}_d(t)\right)\left(\mathbf{d}(t')-\boldsymbol{\mu}_d(t')\right)^T\right]\\
&= \int_0^{t'}\int_0^{t}\int_0^{\tau'}\int_0^{\tau} E\left[\left(\mathbf{a}(s)-\boldsymbol{\mu}_a(s)\right)\left(\mathbf{a}(s')-\boldsymbol{\mu}_a(s')\right)^T\right] ds\,ds'\,d\tau\,d\tau'\\
&\quad+\int_0^{t'}\int_0^{t}\int_0^{\tau} E\left[\left(\mathbf{a}(s)-\boldsymbol{\mu}_a(s)\right)\left(\mathbf{v}(0)-\boldsymbol{\mu}_v(0)\right)^T\right] ds\,d\tau\,d\tau'\\
&\quad+\int_0^{t'}\int_0^{\tau'}\int_0^{t} E\left[\left(\mathbf{v}(0)-\boldsymbol{\mu}_v(0)\right)\left(\mathbf{a}(s')-\boldsymbol{\mu}_a(s')\right)^T\right] d\tau\,ds'\,d\tau'\\
&\quad+\int_0^{t}\int_0^{\tau} E\left[\left(\mathbf{a}(s)-\boldsymbol{\mu}_a(s)\right)\left(\mathbf{d}(0)-\boldsymbol{\mu}_d(0)\right)^T\right] ds\,d\tau\\
&\quad+\int_0^{t'}\int_0^{\tau'} E\left[\left(\mathbf{d}(0)-\boldsymbol{\mu}_d(0)\right)\left(\mathbf{a}(s')-\boldsymbol{\mu}_a(s')\right)^T\right] ds'\,d\tau'\\
&\quad+\int_0^{t'}\int_0^{t} E\left[\left(\mathbf{v}(0)-\boldsymbol{\mu}_v(0)\right)\left(\mathbf{v}(0)-\boldsymbol{\mu}_v(0)\right)^T\right] d\tau\,d\tau'\\
&\quad+\int_0^{t} E\left[\left(\mathbf{v}(0)-\boldsymbol{\mu}_v(0)\right)\left(\mathbf{d}(0)-\boldsymbol{\mu}_d(0)\right)^T\right] d\tau\\
&\quad+\int_0^{t'} E\left[\left(\mathbf{d}(0)-\boldsymbol{\mu}_d(0)\right)\left(\mathbf{v}(0)-\boldsymbol{\mu}_v(0)\right)^T\right] d\tau'\\
&\quad+E\left[\left(\mathbf{d}(0)-\boldsymbol{\mu}_d(0)\right)\left(\mathbf{d}(0)-\boldsymbol{\mu}_d(0)\right)^T\right].\end{aligned} \tag{A.14}$$

Once again, we assume the acceleration and initial position and velocity terms are independent, which results in

$$\begin{aligned}\mathcal{K}_{dd}(t,t') &= \int_0^{t'}\int_0^{t}\int_0^{\tau'}\int_0^{\tau} \mathcal{K}_{aa}(s,s')ds\,ds'\,d\tau\,d\tau' + \int_0^{t'}\int_0^{t} \mathcal{K}_{vv}(0,0)d\tau\,d\tau' \\ &\quad + \int_0^{t} \mathcal{K}_{vd}(0,0)d\tau + \int_0^{t'} \mathcal{K}_{dv}(0,0)d\tau' + \mathcal{K}_{dd}(0,0) \\ &= \int_0^{t'}\int_0^{t}\int_0^{\tau'}\int_0^{\tau} \mathcal{K}_{aa}(s,s')ds\,ds'\,d\tau\,d\tau' + \mathcal{K}_{vv}(0,0)tt' \\ &\quad + \mathcal{K}_{vd}(0,0)t + \mathcal{K}_{dv}(0,0)t' + \mathcal{K}_{dd}(0,0). \end{aligned} \tag{A.15}$$

where $\mathcal{K}_{vd}(0,0)$ and $\mathcal{K}_{dv}(0,0)$ are the position-velocity cross-covariance initial uncertainties, and $\mathcal{K}_{vv}(0,0)$ and $\mathcal{K}_{dd}(0,0)$ are the initial velocity and position uncertainties, respectively.

For completeness, we obtain the cross-covariance functions in a similar manner:

$$\mathcal{K}_{vd}(t,t') = \int_0^{t}\int_0^{t'}\int_0^{\tau'} \mathcal{K}_{aa}(\tau,s')ds'\,d\tau'\,d\tau + \mathcal{K}_{vv}(0,0)t' + \mathcal{K}_{vd}(0,0), \tag{A.16}$$

$$\mathcal{K}_{dv}(t,t') = \int_0^{t'}\int_0^{t}\int_0^{\tau} \mathcal{K}_{aa}(s,\tau')ds\,d\tau\,d\tau' + \mathcal{K}_{vv}(0,0)t + \mathcal{K}_{dv}(0,0). \tag{A.17}$$

A.1.2 Specific Example of a White Noise Prior on $\ddot{\mathbf{x}}(t)$

We develop the specific GP used in this thesis by starting with the continuous white noise acceleration (CWNA) (Bar-Shalom et al., 2002) model. That is,

$$\ddot{\mathbf{x}}(t) \sim \mathcal{GP}(\mathbf{0}, \mathbf{W}\,\delta(t-t')), \tag{A.18}$$

where $\mathbf{W}$ is a power spectral density matrix, and $\delta(\cdot)$ is the Dirac delta function. Once again, this can be interpreted as a constant velocity assumption with perturbative noise. If $\mathbf{v}(0) = \mathbf{0}$ and $\mathbf{d}(0) = \mathbf{0}$, it is straightforward to show that the mean functions for position, velocity, and acceleration are all zero. As a result, only the covariance functions are derived in the following sections.

A.1.2.1 Acceleration

In the case of acceleration, the covariance function is simply as stated above:

$$\mathcal{K}_{aa}(t,t') = \mathbf{W}\,\delta(t-t'). \tag{A.19}$$

A.1.2.2 Velocity

Next, we integrate the acceleration covariance function to obtain

$$\begin{aligned}\int_0^{t'}\int_0^{t} \boldsymbol{\mathcal{K}}_{aa}(\tau,\tau')d\tau\, d\tau' &= \mathbf{W}\int_0^{t'}\int_0^{t}\delta(\tau-\tau')d\tau\, d\tau' \\ &= \mathbf{W}\int_0^{t'}\left(H(t-\tau')-H(0-\tau')\right)d\tau' \\ &= \mathbf{W}\int_0^{t'} H(t-\tau')d\tau', \qquad \text{(A.20)}\end{aligned}$$

where $H(t)$ is the Heaviside step function. We now consider the two cases. For $t \geq t'$,

$$\mathbf{W}\int_0^{t'} H(t-\tau')d\tau' = \mathbf{W}\int_0^{t'} d\tau' = \mathbf{W}t', \qquad \text{(A.21)}$$

and for $t < t'$,

$$\mathbf{W}\int_0^{t'} H(t-\tau')d\tau' = \mathbf{W}\int_0^{t} d\tau' = \mathbf{W}t. \qquad \text{(A.22)}$$

Inserting this result into (A.11) provides the velocity covariance function, which can be summarized by

$$\boldsymbol{\mathcal{K}}_{vv}(t,t') = \mathbf{W}\min(t,t') + \boldsymbol{\mathcal{K}}_{vv}(0,0). \qquad \text{(A.23)}$$

A.1.2.3 Position

To obtain the position covariance function, we integrate the expression from the previous section two more times:

$$\int_0^{t'}\int_0^{t}\int_0^{\tau'}\int_0^{\tau} \boldsymbol{\mathcal{K}}_{aa}(s,s')ds\, ds'\, d\tau\, d\tau' = \int_0^{t'}\int_0^{t}\mathbf{W}\min(\tau,\tau')d\tau\, d\tau'. \qquad \text{(A.24)}$$

We approach this integral in a similar manner by considering the two cases separately. For $t \geq t'$, we obtain

$$\begin{aligned}\int_0^{t'}\int_0^{t}\mathbf{W}\min(\tau,\tau')d\tau\, d\tau' &= \mathbf{W}\int_0^{t'}\left(\int_0^{\tau'}\min(\tau,\tau')d\tau + \int_{\tau'}^{t}\min(\tau,\tau')d\tau\right)d\tau' \\ &= \mathbf{W}\int_0^{t'}\left(\int_0^{\tau'}\tau d\tau + \int_{\tau'}^{t}\tau' d\tau\right)d\tau' \\ &= \mathbf{W}\int_0^{t'}\left(\frac{\tau'^2}{2}+\tau'(t-\tau')\right)d\tau' \\ &= \mathbf{W}\int_0^{t'}\left(t\tau'-\frac{\tau'^2}{2}\right)d\tau' \\ &= \mathbf{W}\left(\frac{tt'^2}{2}-\frac{t'^3}{6}\right), \qquad \text{(A.25)}\end{aligned}$$

and for $t < t'$,

$$\begin{aligned}\int_0^{t'}\int_0^{t} \mathbf{W}\min(\tau,\tau')d\tau\, d\tau' &= \mathbf{W}\left(\int_0^{t}\int_0^{t}\min(\tau,\tau')d\tau\, d\tau' + \int_t^{t'}\int_0^{t}\min(\tau,\tau')d\tau\, d\tau'\right)\\ &= \mathbf{W}\left(\frac{t^3}{2} - \frac{t^3}{6} + \int_t^{t'}\int_0^{t}\tau d\tau\, d\tau'\right)\\ &= \mathbf{W}\left(\frac{t^3}{3} + \int_t^{t'}\frac{t^2}{2}d\tau'\right)\\ &= \mathbf{W}\left(\frac{t^3}{3} + (t'-t)\frac{t^2}{2}\right)\\ &= \mathbf{W}\left(\frac{t't^2}{2} - \frac{t^3}{6}\right). \qquad (A.26)\end{aligned}$$

This expression can then be inserted into (A.15) and simplified. As a result, the position covariance function can be expressed as

$$\begin{aligned}\boldsymbol{\mathcal{K}}_{dd}(t,t') = \mathbf{W}\left(\frac{\min(t,t')^2\max(t,t')}{2} - \frac{\min(t,t')^3}{6}\right) + \boldsymbol{\mathcal{K}}_{vv}(0,0)tt'\\ + \boldsymbol{\mathcal{K}}_{vd}(0,0)t + \boldsymbol{\mathcal{K}}_{dv}(0,0)t' + \boldsymbol{\mathcal{K}}_{dd}(0,0). \qquad (A.27)\end{aligned}$$

A.1.2.4 Summary

In summary, the covariance functions for white noise on acceleration case are

$$\boldsymbol{\mathcal{K}}_{aa}(t,t') = \mathbf{W}\,\delta(t-t'), \qquad (A.28)$$

$$\boldsymbol{\mathcal{K}}_{vv}(t,t') = \mathbf{W}\min(t,t') + \boldsymbol{\mathcal{K}}_{vv}(0,0), \qquad (A.29)$$

$$\begin{aligned}\boldsymbol{\mathcal{K}}_{dd}(t,t') = \mathbf{W}\left(\frac{\min(t,t')^2\max(t,t')}{2} - \frac{\min(t,t')^3}{6}\right) + \boldsymbol{\mathcal{K}}_{vv}(0,0)tt'\\ + \boldsymbol{\mathcal{K}}_{vd}(0,0)t + \boldsymbol{\mathcal{K}}_{dv}(0,0)t' + \boldsymbol{\mathcal{K}}_{dd}(0,0), \qquad (A.30)\end{aligned}$$

and the cross-covariance functions (with the derivations left to the reader) are

$$\boldsymbol{\mathcal{K}}_{vd}(t,t') = \mathbf{f}(t,t') + \boldsymbol{\mathcal{K}}_{vv}(0,0)t' + \boldsymbol{\mathcal{K}}_{vd}(0,0), \qquad (A.31)$$

$$\boldsymbol{\mathcal{K}}_{dv}(t,t') = \mathbf{f}(t',t) + \boldsymbol{\mathcal{K}}_{vv}(0,0)t + \boldsymbol{\mathcal{K}}_{dv}(0,0), \qquad (A.32)$$

$$\mathbf{f}(\tau,\tau') = \mathbf{W}\begin{cases}\tau\tau' - \frac{\tau^2}{2} & ,\tau < \tau'\\ \frac{\tau'^2}{2} & ,\tau \geq \tau'\end{cases}. \qquad (A.33)$$

A.2 Incorporating Derivative-Type Measurements

In this appendix, we detail how to handle derivative-type measurements in GPGN. This is utilized during the experimental validation in Chapter 3 to incorporate instantaneous linear and angular velocity measurements obtained from wheel odometry. Derivative-type measurements are incorporated in a simple manner using the weight-space approach. Measurements of the form

$$\mathbf{v}_k = \mathbf{g}_k(\dot{\boldsymbol{\theta}}(t_k)) + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k), \tag{A.34}$$

are linearized by first approximating the state derivative by the current guess, $\dot{\bar{\boldsymbol{\theta}}}(t)$, and an additive perturbation, $\delta\dot{\boldsymbol{\theta}}(t)$. By applying the basis function representation $\dot{\boldsymbol{\theta}}(t) = \dot{\boldsymbol{\Psi}}(t)\,\mathbf{b}$, we get

$$\begin{aligned} \dot{\boldsymbol{\theta}}(t) &\approx \dot{\bar{\boldsymbol{\theta}}}(t) + \delta\dot{\boldsymbol{\theta}}(t) \\ &= \dot{\boldsymbol{\Psi}}(t)\,\bar{\mathbf{b}} + \dot{\boldsymbol{\Psi}}(t)\,\delta\mathbf{b}, \end{aligned} \tag{A.35}$$

which results in

$$\mathbf{v}_k \approx \mathbf{g}_k(\dot{\bar{\boldsymbol{\theta}}}(t_k)) + \mathbf{G}_k \dot{\boldsymbol{\Psi}}(t_k)\,\delta\mathbf{b} + \mathbf{w}_k, \quad \mathbf{G}_k := \left.\frac{\partial \mathbf{g}_k}{\partial \dot{\boldsymbol{\theta}}}\right|_{\dot{\bar{\boldsymbol{\theta}}}(t_k)}. \tag{A.36}$$

Given a total of K derivative measurements, we stack them along with the other measurements by redefining the (3.20) matrices to be

$$\mathbf{z} := \begin{bmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_N \\ \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_K \end{bmatrix}, \quad \mathbf{h} := \begin{bmatrix} \mathbf{h}_1(\bar{\boldsymbol{\theta}}(t_1)) \\ \vdots \\ \mathbf{h}_N(\bar{\boldsymbol{\theta}}(t_N)) \\ \mathbf{g}_1(\dot{\bar{\boldsymbol{\theta}}}(t_1)) \\ \vdots \\ \mathbf{g}_K(\dot{\bar{\boldsymbol{\theta}}}(t_K)) \end{bmatrix}, \quad \boldsymbol{\Psi} := \begin{bmatrix} \boldsymbol{\Phi} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}, \quad \boldsymbol{\Phi} := \begin{bmatrix} \boldsymbol{\Phi}(t_1) \\ \vdots \\ \boldsymbol{\Phi}(t_N) \\ \dot{\boldsymbol{\Phi}}(t_1) \\ \vdots \\ \dot{\boldsymbol{\Phi}}(t_K) \end{bmatrix},$$

$$\mathbf{R} := \begin{bmatrix} \mathbf{R}_1 & & & & & \\ & \ddots & & & & \\ & & \mathbf{R}_N & & & \\ & & & \mathbf{Q}_1 & & \\ & & & & \ddots & \\ & & & & & \mathbf{Q}_K \end{bmatrix}, \quad \mathbf{H} := \begin{bmatrix} \mathbf{H}_1 & & & & & \\ & \ddots & & & & \\ & & \mathbf{H}_N & & & \\ & & & \mathbf{G}_1 & & \\ & & & & \ddots & \\ & & & & & \mathbf{G}_K \end{bmatrix}, \tag{A.37}$$

and the derivation continues as before.

In the final step of the weight-space derivation, the $\mathbf{\Phi K \Phi}^T$ terms are replaced by evaluating the covariance function in (3.32). A similar substitution occurs when incorporating derivative measurements, but the basis function derivatives, $\dot{\mathbf{\Phi}}(t)$, are now involved. The GPGN algorithm can be recovered by redefining the $\boldsymbol{\mathcal{K}}$ matrices appropriately using the substitutions

$$\mathbf{\Phi}(t)\,\mathbf{K\Phi}(t')^T = \boldsymbol{\mathcal{K}}(t,t'), \tag{A.38a}$$

$$\dot{\mathbf{\Phi}}(t)\,\mathbf{K\Phi}(t')^T = \frac{\partial\,\boldsymbol{\mathcal{K}}(t,t')}{\partial t}, \tag{A.38b}$$

$$\mathbf{\Phi}(t)\,\mathbf{K}\dot{\mathbf{\Phi}}(t')^T = \frac{\partial\,\boldsymbol{\mathcal{K}}(t,t')}{\partial t'}, \tag{A.38c}$$

$$\dot{\mathbf{\Phi}}(t)\,\mathbf{K}\dot{\mathbf{\Phi}}(t')^T = \frac{\partial^2\,\boldsymbol{\mathcal{K}}(t,t')}{\partial t \partial t'}. \tag{A.38d}$$

Furthermore, these derivative expressions are related to the covariance functions developed in Appendix A by

$$\boldsymbol{\mathcal{K}}_{dd}(t,t') = \boldsymbol{\mathcal{K}}(t,t'), \tag{A.39a}$$

$$\boldsymbol{\mathcal{K}}_{vd}(t,t') = \frac{\partial\,\boldsymbol{\mathcal{K}}(t,t')}{\partial t}, \tag{A.39b}$$

$$\boldsymbol{\mathcal{K}}_{dv}(t,t') = \frac{\partial\,\boldsymbol{\mathcal{K}}(t,t')}{\partial t'}, \tag{A.39c}$$

$$\boldsymbol{\mathcal{K}}_{vv}(t,t') = \frac{\partial^2\,\boldsymbol{\mathcal{K}}(t,t')}{\partial t \partial t'}. \tag{A.39d}$$

These expressions can also be utilized in the linear prediction equations (3.49)-(3.50), by replacing $\boldsymbol{\mathcal{K}}(t)$ appropriately to obtain derivative state values. Instantaneous acceleration measurements and state predictions can be incorporated in a similar manner.

Appendix B

Normalized Innovation Squared Test

This appendix provides a derivation of the Normalized Innovation Squared (NIS) test (Bar-Shalom et al., 2002), along with the adaptation for heterogeneous measurements described in Tong and Barfoot (2013a). This test is employed for heterogeneous measurement outlier rejection in Chapter 5. For outlier rejection, we consider the problem where we are given a set of measurements and a hypothesis for the state of the world, and we must determine which of the measurements agree with the hypothesis. We begin by determining the system models, and conclude with the derivation of the NIS test.

B.1 Measurement Models

For clarity in developing the statistical test, we start by defining the form of the measurement models for the odometry and the feature position measurements. We begin by observing the fact that both can be expressed in a common probabilistic form. That is, if we define $\boldsymbol{\theta}$ to be the true state, we can express both as

$$\mathbf{z}_i := \mathbf{h}_i(\boldsymbol{\theta}) + \mathbf{n}_i, \quad \mathbf{n}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_i). \tag{B.1}$$

This is the same generic measurement model utilized in Chapters 2 and 3. Though we present a vector space derivation for clarity, this test is easily adapted to 3D state representations using the methods described in Chapter 4. Since most measurement types can be expressed in this common form, other than the size of the vectors involved, there is no need to differentiate between measurement types.

B.2 Hypothesis Model

In the outlier rejection problem, we are tasked with detecting outliers in a set of measurements, as compared to a hypothesis model. This hypothesis model consists of some estimated values of the state, $\overline{\boldsymbol{\theta}}$, as well as the uncertainties associated to them. We model this uncertainty as a Gaussian random variable with the distribution

$$\overline{\boldsymbol{\theta}} := \boldsymbol{\theta} + \delta\boldsymbol{\theta}, \quad \delta\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \overline{\boldsymbol{\Sigma}}), \tag{B.2}$$

where we assume the estimated state models the true state, with an additive error, $\delta\boldsymbol{\theta}$. This additive error is zero-mean, and Gaussian-distributed with covariance $\overline{\boldsymbol{\Sigma}}$, that relates to our confidence in the estimate.

B.3 Statistical Test

The first step in the statistical test is to compute the estimated error for each measurement, $\overline{\mathbf{e}}_i$, which is the difference between the measurement obtained by the sensor and the expected measurement computed using the hypothesis model.

$$\overline{\mathbf{e}}_i := \mathbf{z}_i - \mathbf{h}_i(\overline{\boldsymbol{\theta}}). \tag{B.3}$$

We term this the *estimated error* because the true state is unavailable, so the measurement error is *estimated* based on the hypothesis model. In the standard test that is utilized by RANSAC (Fischler and Bolles, 1981), the estimated error is assumed to follow the same probability distribution as the true error. That is,

$$\overline{\mathbf{e}}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_i). \tag{B.4}$$

However, this standard approach does not take into account the fact that there is also an uncertainty to the hypothesis model. This can lead to many inliers misclassified as outliers. For example, if we are less certain about a rover pose and feature position, we should also be less certain about the exact distance between them. Therefore, to compute the proper threshold for outlier classification, the hypothesis model uncertainty should be propagated through the measurement error model. One method of doing this is to linearize the estimated error model, and pass the covariance through the linear system.

Let us define the linearized form of the estimated error (B.3) as

$$\begin{aligned}\overline{\mathbf{e}}_i &= \mathbf{z}_i - \mathbf{h}_i(\overline{\boldsymbol{\theta}}) \\ &\approx \mathbf{z}_i - \mathbf{h}_i(\boldsymbol{\theta}) - \overline{\mathbf{H}}_i\,\delta\boldsymbol{\theta} \\ &= \mathbf{e}_i - \overline{\mathbf{H}}_i\,\delta\boldsymbol{\theta}, \end{aligned} \tag{B.5}$$

$$\overline{\mathbf{H}}_i := \left.\frac{\partial \mathbf{h}_i(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\overline{\boldsymbol{\theta}}}, \tag{B.6}$$

where $\overline{\mathbf{H}}_i$ is the Jacobian of the ith measurement function evaluated at $\overline{\boldsymbol{\theta}}$. Though a standard linearization would evaluate the Jacobian at $\boldsymbol{\theta}$, in estimation, the true state is unavailable. Using the measurement model given by (B.1), we can analyze the statistical properties of $\overline{\mathbf{e}}_i$. For the first moment, we get

$$\begin{aligned}E\left[\overline{\mathbf{e}}_i\right] &= E\left[\mathbf{e}_i - \overline{\mathbf{H}}_i\,\delta\boldsymbol{\theta}\right] \\ &= E\left[\mathbf{e}_i\right] - \overline{\mathbf{H}}_i\,E\left[\delta\boldsymbol{\theta}\right] \\ &= \mathbf{0}, \end{aligned} \tag{B.7}$$

and for the second moment, we have

$$\begin{aligned}E\left[\overline{\mathbf{e}}_i\,\overline{\mathbf{e}}_i^T\right] &= E\left[\left(\mathbf{e}_i - \overline{\mathbf{H}}_i\,\delta\boldsymbol{\theta}\right)\left(\mathbf{e}_i - \overline{\mathbf{H}}_i\,\delta\boldsymbol{\theta}\right)^T\right] \\ &= E\left[\mathbf{e}_i\,\mathbf{e}_i^T\right] - E\left[\mathbf{e}_i\,\delta\boldsymbol{\theta}^T\,\overline{\mathbf{H}}_i^T\right] - E\left[\overline{\mathbf{H}}_i\,\delta\boldsymbol{\theta}\,\mathbf{e}_i^T\right] + E\left[\overline{\mathbf{H}}_i\,\delta\boldsymbol{\theta}\,\delta\boldsymbol{\theta}^T\,\overline{\mathbf{H}}_i^T\right]. \end{aligned} \tag{B.8}$$

Since the hypothesis model is separate from the measurements we wish to classify, the hypothesis model error, $\delta\boldsymbol{\theta}$, and the measurement errors, $\mathbf{e}_i$, are independent. Applying this fact, that the expected value of the cross terms should be zero, results in a simple expression for the propagated uncertainty of the estimated errors:

$$\begin{aligned}E\left[\overline{\mathbf{e}}_i\,\overline{\mathbf{e}}_i^T\right] &= E\left[\mathbf{e}_i\,\mathbf{e}_i^T\right] + E\left[\overline{\mathbf{H}}_i\,\delta\boldsymbol{\theta}\,\delta\boldsymbol{\theta}^T\,\overline{\mathbf{H}}_i^T\right] \\ &= \mathbf{R}_i + \overline{\mathbf{H}}_i\,E\left[\delta\boldsymbol{\theta}\,\delta\boldsymbol{\theta}^T\right]\overline{\mathbf{H}}_i^T \\ &= \mathbf{R}_i + \overline{\mathbf{H}}_i\,\overline{\boldsymbol{\Sigma}}\,\overline{\mathbf{H}}_i^T. \end{aligned} \tag{B.9}$$

Finally, by maintaining the Gaussian noise assumption, we can summarize the noise properties of $\overline{\mathbf{e}}_i$ as

$$\overline{\mathbf{e}}_i \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}_i + \overline{\mathbf{H}}_i\,\overline{\boldsymbol{\Sigma}}\,\overline{\mathbf{H}}_i^T\right). \tag{B.10}$$

As can be seen, the covariance of the estimated error is inflated by the uncertainty in the hypothesis model. Although the use of a hypothesis model actually couples the estimated errors together, the NIS test retains the assumption of measurement independence for computational efficiency. As demonstrated in Tong and Barfoot (2013a), the NIS test scales linearly with the number of measurements, while a test that considers the coupling between measurements in a batch approach scales in a quadratic manner.

Using these properties, the Mahalanobis distance (Mahalanobis, 1936) can be computed for each estimated error. The distances associated to inlier measurements should follow a chi-squared distribution with a degree-of-freedom equal to the dimension of the measurement.

$$\bar{\mathbf{e}}_i^T \left(\mathbf{R}_i + \overline{\mathbf{H}}_i \overline{\boldsymbol{\Sigma}} \overline{\mathbf{H}}_i^T\right)^{-1} \bar{\mathbf{e}}_i \sim \chi^2_{\dim(\bar{\mathbf{e}}_i)}. \tag{B.11}$$

Therefore, the NIS test is as follows: if the value of a Mahalonobis distance exceeds a predefined threshold (usually corresponding to the 3σ boundary for a one-dimensional Gaussian distribution, or 99.7% of the probability density), the measurement is classified as an outlier. This test is repeated for each measurement, producing a set of outlier classifications.

Bibliography

Anderson, S. and Barfoot, T. D. (2013a). RANSAC for motion-distorted 3D visual sensors. Submitted to the *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Anderson, S. and Barfoot, T. D. (2013b). Towards relative continuous-time SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1025–1032, Karlsruhe, Germany.

Anderson, S., McManus, C., Dong, H., Beerepoot, E., and Barfoot, T. D. (2013). The gravel pit lidar intensity imagery dataset. Submitted to the *International Journal of Robotics Research*.

Archambeau, C., Cornford, D., Opper, M., and Shawe-Taler, J. (2007). Gaussian process approximations of stochastic differential equations. *Journal of Machine Learning Research: Workshop and Conference Proceedings*, 1:1–16.

Bailey, T., Neito, J., Guivant, J., Stevens, M., and Nebot, E. (2006). Consistency of the EKF-SLAM algorithm. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3562–3568, Beijing, China.

Bakambu, J., Allard, P., and Dupuis, E. (2006a). 3D terrain modeling for rover localization and navigation. In *Proceedings of the 3rd Canadian Conference on Computer and Robot Vision*, pages 61–68, Quebec City, Quebec, Canada.

Bakambu, J., Gemme, S., and Dupuis, E. (2006b). Rover localization through 3D terrain registration in natural environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2308–2313, Beijing, China.

Bar-Shalom, Y., Kirubarajan, T., and Li, X. (2002). *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., New York, NY, USA.

Barfoot, T. D., Forbes, J. R., and Furgale, P. T. (2011). Pose estimation using linearized rotations and quaternion algebra. *Acta Astronautica*, 68(1-2):101–112.

Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V. (2008). SURF: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359.

Bell, B. M. and Cathey, F. W. (1993). The iterated Kalman filter update as a Gauss-Newton method. *IEEE Transactions on Automatic Control*, 38(2):294–297.

Besl, P. and McKay, H. (1992). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256.

Bibby, C. and Reid, I. D. (2010). A hybrid SLAM representation for dynamic marine environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 257–264, Anchorage, AK, USA.

Borrmann, D., Elseberg, J., Lingemann, K., Nüchter, A., and Hertzberg, J. (2008). Globally consistent 3D mapping with scan matching. *Robotics and Autonomous Systems*, 56(2):130–142.

Bosse, M. and Zlot, R. (2009). Continuous 3D scan-matching with a spinning 2D laser. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, pages 4312–4319, Kobe, Japan.

Bosse, M., Zlot, R., and Flick, P. (2012). Zebedee: Design of a spring-mounted 3-D range sensor with application to mobile mapping. *IEEE Transactions on Robotics*, 28(5):1104–1119.

Brenner, C., Dold, C., and Ripperda, N. (2008). Coarse orientation of terrestrial laser scans in urban environments. *ISPRS Journal of Photogrammetry & Remote Sensing, Special Issue on "Terrestrial Laser Scanning"*, 63(1):4–18.

Brown, D. C. (1958). A solution to the general problem of multiple station analytical stereotriangulation. Technical Report 43, Patrick Airforce Base, Florida, USA.

Carle, P. J. F., Furgale, P. T., and Barfoot, T. D. (2010). Long-range rover localization by matching lidar scans to orbital elevation maps. *Journal of Field Robotics*, 27(3):344–370.

Chen, C., Hung, H., and Cheng, J. (1999). RANSAC-based DARCES: A new approach to fast automatic registration of partially overlapping range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1229–1234.

Chen, C. and Medioni, G. (1992). Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155.

Chli, M. and Davison, A. J. (2009). Active matching for visual tracking. *Robotics and Autonomous Systems*, 57(12):1173–1187.

Colaprete, A., Schultz, P., Heldmann, J., Wooden, D., Shirley, M., Ennico, K., Hermalyn, B., Marshall, W., Ricco, A., Elphic, R. C., Goldstein, D., Summy, D., Bart, G. D., Asphaug, E., Korycansky, D., Landis, D., and Sollitt, L. (2010). Detection of water in the LCROSS ejecta plume. *Science*, 330(6003):463–468.

Cozman, F., Krotkov, E., and Guestrin, C. (2000). Outdoor visual position estimation for planetary rovers. *Autonomous Robots*, 9(2):135–150.

Cunnington, S. J. and Stoddart, A. J. (1999). N-view point set registration: A comparison. In *Proceedings of the 10th British Machine Vision Conference*, pages 234–244, Nottingham, UK.

Dale, S. (2006). Exploration strategy and architecture. In *Implementing the Vision - 2nd Space Exploration Conference*, Houston, Texas, USA.

Deisenroth, M. P., Turner, R., Huber, M. F., Hanebeck, U. D., and Rasmussen, C. E. (2012). Robust filtering and smoothing with gaussian processes. *IEEE Transactions on Automatic Control*, 57(7):1865–1871.

Dellaert, F. and Kaess, M. (2006). Square root SAM: Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research (IJRR)*, 25(12):1181–1203.

Dong, H., Anderson, S., and Barfoot, T. D. (2013). Two-axis scanning lidar geometric calibration using intensity imagery and distortion mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3657–3663, Karlsruhe, Germany.

Dong, H. and Barfoot, T. D. (2012). Lighting-invariant visual odometry using lidar intensity imagery and pose interpolation. In *Proceedings of the International Conference on Field and Service Robotics (FSR)*, Matsushima, Japan.

Fairfield, N., Wettergreen, D., and Kantor, G. (2010). Segmented SLAM in three-dimensional environments. *Journal of Field Robotics*, 27(1):85–103.

Ferris, B., Fox, D., and Lawrence, N. (2007). Wifi-SLAM using Gaussian process latent variable models. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2480–2485.

Ferris, B., Hähnel, D., and Fox, D. (2006). Gaussian processes for signal strength-based localization. In *Proceedings of Robotics: Science and Systems (RSS)*, Philadelphia, USA.

Fischler, M. and Bolles, R. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

Fong, T., Allan, M., Bouyssounouse, X., Bualat, M. G., Deans, M. C., Edwards, L., Flückiger, L., Keely, L., Lee, S. Y., Lees, D., To, V., and Utz, H. (2008a). Robotic site survey at haughton crater. In *Proceedings of the 9th International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, Los Angeles, CA, USA.

Fong, T., Bualat, M., Deans, M., Allan, M., Bouyssounouse, X., Broxton, M., Edwards, L., Elphic, R., Flückiger, L., Frank, J., Keely, L., Kobayashi, L., Lee, P., Lee, S. Y., Lees, D., Pacis, E., Park, E., Pedersen, L., Schreckenghost, D., Smith, T., To, V., and

Utz, H. (2008b). Field testing of utility robots for lunar surface operations. In *Space 2008 Conference and Exhibit*, San Diego, CA, USA.

Furgale, P., Barfoot, T., and Sibley, G. (2012). Continuous-time batch estimation using temporal basis functions. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2088–2095, St. Paul, MN, USA.

Furgale, P. T. (2011). *Extensions to the Visual Odometry Pipeline for the Exploration of Planetary Surfaces*. PhD thesis, University of Toronto.

Furgale, P. T. and Barfoot, T. D. (2010). Visual teach and repeat for long-range rover autonomy. *Journal of Field Robotics, Special issue on "Visual mapping and navigation outdoors"*, 27(5):534–560.

Furgale, P. T. and Barfoot, T. D. (2013). Associating uncertainty with three-dimensional poses for use in estimation problems. Submitted to the *IEEE Transactions on Robotics*.

Gauss, C. F. (1855). *Méthode des Moindres Carrés*. Mallet-Bachelier, Impreimeur-Libraire de L'École Polytechnique, Quai des Augustins no. 55, Paris.

Gemme, S., Gingras, D., Salerno, A., Dupuis, E., Pomerleau, F., and Michaud, F. (2012). Pose refinement using ICP applied to 3-D LIDAR data for exploration rovers. In *Proceedings of the 11th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, Turin, Italy.

Guivant, J., Nebot, E., and Baiker, S. (2000). Localization and map building using laser range sensors in outdoor applications. *Journal of Robotic Systems*, 17:565–583.

Guizilini, V. and Ramos, F. (2012). Semi-parametric models for visual odometry. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3482–3489, St. Paul, MN, USA.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the Fourth Alvey Vision Conference*, pages 147–151, Manchester, UK.

Hartley, R. I. and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049.

Hedborg, J., Forssén, P.-E., Felsberg, M., and Ringaby, E. (2012). Rolling shutter bundle adjustment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1434–1441, Providence, RI, USA.

Heider, P., Pierre-Pierre, A., Li, R., and Grimm, C. (2011). Local shape descriptors, a survey and evaluation. In *Eurographics Workshop on 3D Object Retrieval*, pages 49–57, Llandudno, UK.

Huber, P. J. (1981). *Robust Statistics*. John Wiley & Sons, New York, NY, USA.

Hughes, P. C. (1986). *Spacecraft Attitude Dynamics*. John Wiley & Sons Inc, New York, NY, USA.

Husmann, K. and Pedersen, L. (2008). Strobe lit high dynamic range stereo imagery for dark navigation. In *Proceedings of the 9th International Symposium on Artificial Intelligence for Robotics and Automation (iSAIRAS)*, Los Angeles, CA, USA.

Jazwinski, A. H. (1970). *Stochastic Processes and Filtering Theory.* New York: Academic Press.

Johnson, A. (1997). *Spin-images: A representation for 3-D surface matching.* PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA.

Johnson, A. E., Goldberg, S. B., Cheng, Y., and Matthies, L. H. (2008). Robust and efficient stereo feature tracking for visual odometry. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 39–46, Pasadena, CA, USA.

Johnson-Roberson, M., Pizarro, O., Williams, S. B., and Mahon, I. (2010). Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys. *Journal of Field Robotics*, 27(1):21–51.

Julier, S. J. and Uhlmann, J. K. (1997). A new extension of the Kalman Filter to nonlinear systems. In *AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, Orlando, FL, USA.

Julier, S. J. and Uhlmann, J. K. (2001). A counter example to the theory of simultaneous localization and map building. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 4238–4243, Seoul, Korea.

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J., and Dellaert, F. (2012). iSAM2: Incremental smoothing and mapping using the Bayes tree. *International Journal of Robotics Research (IJRR)*, 31(2):216–235.

Kaess, M., Ranganathan, A., and Dellaert, F. (2008). iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45.

Kalman, R. E. and Bucy, R. S. (1961). New results in linear filtering and prediction theory. *Journal of Basic Engineering*, 83(1):95–108.

Kelly, J. and Sukhatme, G. S. (2011). Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *International Journal of Robotics Research (IJRR)*, 30(1):56–79.

King, B. J., Malisiewicz, T., Stewart, C. V., and Radke, R. J. (2005). Registration of multiple range scans as a location recognition problem: Hypothesis generation, refinement and verification. In *Proceedings of the International Conference on 3-D Digital Imaging and Modeling (3DIM)*, pages 180–187, Ottawa, Ontario, Canada.

Ko, J. and Fox, D. (2009). GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models. *Autonomous Robots*, 27(1):75–90.

Ko, J. and Fox, D. (2011). Learning GP-BayesFilters via Gaussian process latent variable models. *Autonomous Robots*, 30(1):3–23.

Konolige, K. and Agrawal, M. (2008). FrameSLAM: From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics*, 24(5):1066–1077.

Konolige, K., Agrawal, M., and Sola, J. (2007). Large scale visual odometry for rough terrain. In *Proceedings of the International Symposium on Robotics Research*, pages 201–212, Hiroshima, Japan.

Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., and Burgard, W. (2011). g^2o: A general framework for graph optimization. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 3607–3613, Shanghai, China.

Kümmerle, R., Steder, B., Dornhege, C., Ruhnke, M., Grisetti, G., Stachniss, C., and Kleiner, A. (2009). On measuring the accuracy of SLAM algorithms. *Autonomous Robots, Special Issue on "Characterizing Mobile Robot Localization and Mapping"*, 27(4):387–407.

Lawrence, N. (2003). Gaussian process latent variable models for visualization of high dimensional data. In *Advances in Neural Information Processing Systems (NIPS)*.

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *The Quarterly of Applied Mathematics*, 2:164–168.

Levinson, J. (2011). *Automatic laser calibration, mapping, and location for autonomous vehicles*. PhD thesis, Stanford University.

Li, R., Di, K., Howard, A. B., Matthies, L., Wang, J., and Agarwal, S. (2007). Rock modeling and matching for autonomous long-range mars rover localization. *Journal of Field Robotics*, 24(3):187–203.

Liebe, C. C. (1995). Star trackers for attitude determination. *IEEE Aerospace and Electronic Systems Magazine*, 10(6):10–16.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Lu, F. and Milios, E. (1994). Robot pose estimation in unknown environments by matching 2D range scans.

Magnusson, M., Andreasson, H., Nüchter, A., and Lilienthal, A. J. (2009). Automatic appearance-based loop detection from three-dimensional laser data using the normal distributions transform. *Journal of Field Robotics*, 26(11-12):892–914.

Magnusson, M., Lilienthal, A. J., Andreasson, H., and Duckett, T. (2007). Scan registration for autonomous mining vehicles using 3D-NDT. *Journal of Field Robotics*, 24(10):803–827.

Mahalanobis, P. (1936). On the generalized distance in statistics. *Proceedings of the National Institute of Science*, 2(1):49–55.

Maimone, M., Cheng, Y., and Matthies, L. (2007). Two years of visual odometry on the Mars exploration rovers. *Journal of Field Robotics*, 24(3):169–186.

Marquardt, D. (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441.

Matheron, G. (1963). Principles of geostatistics. *Economic Geology*, 58:1246–1266.

McManus, C., Furgale, P., and Barfoot, T. D. (2011). Towards appearance-based methods for lidar sensors. In *Proceedings of the IEEE Conference on Robotics and Automation (ICRA)*, pages 1930–1935, Shanghai, China.

McManus, C., Furgale, P., Stenning, B., and Barfoot, T. D. (2012). Visual teach and repeat using appearance-based lidar. In *Proccedings of the IEEE Conference on Robotics and Automation (ICRA)*, pages 389–396, St. Paul, MN, USA.

Merali, R. S., Tong, C., Gammell, J., Bakambu, J., Dupuis, E., and Barfoot, T. D. (2012). 3D surface mapping using a semi-autonomous rover: A planetary analog field experiment. In *Proceedings of the 11th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, Turin, Italy.

Mikolajczyk, K. and Schmid, C. (2004). Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86.

Moosmann, F. and Stiller, C. (2011). Velodyne SLAM. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 393–398, Baden-Baden, Germany.

Moreaux, G. (2008). Compactly supported radial covariance functions. *Journal of Geodesy*, 82(7):431–443.

Murray, R. M., Li, Z., and Sastry, S. (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press.

Neal, R. M. (1994). *Bayesian Learning for Neural Networks*. PhD thesis, University of Toronto.

Neira, J. and Tardós, J. (2001). Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897.

Neira, J., Tardos, J., Horn, J., and Schmidt, G. (1999). Fusing range and intensity images for mobile robot localization. *IEEE Transactions on Robotics and Automation*, 15(1).

Nüchter, A., Gutev, S., Borrmann, D., and Elseberg, J. (2011). Skyline-based registration of 3D laser scans. *Journal of Geo-spatial Information Science*, 14(2):85–90.

Nüchter, A., Lingemann, K., Hertzberg, J., and Surmann, H. (2007). 6D SLAM - 3D mapping outdoor environments. *Journal of Field Robotics*, 24(8-9):699–722.

Pathak, K., Birk, A., Vaskevicius, N., Pfingsthorn, M., Schwertfeger, S., and Poppinga, J. (2010). Online three-dimensional SLAM by registration of large planar surface segments and closed-form pose-graph relaxation. *Journal of Field Robotics*, 27(1):52–84.

Pedersen, L., Han, C. S., and Vitus, M. (2008). Dark navigation: Sensing and rover navigation in permanently shadowed lunar craters. In *Proceedings of the 9th International Symposium on Artificial Intelligence for Robotics and Automation (iSAIRAS)*, Los Angeles, CA, USA.

Plagemann, C., Kersting, K., Pfaff, P., and Burgard, W. (2007). Gaussian beam processes: A nonparametric Bayesian measurement model for range finders. In *Proceedings of Robotics: Science and Systems (RSS)*, Atlanta, GA, USA.

Pulli, K. (1999). Multiview registration for large data sets. In *Proceedings of the International Conference on 3D Digital Imaging and Modeling (3DIM)*, pages 160–168, Ottawa, Canada.

Ranganathan, A., Yang, M.-H., and Ho, J. (2011). Online sparse matrix Gaussian process regression and vision applications. *IEEE Transactions on Image Processing*, 20(2):391–404.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Rusinkiewicz, S. and Levoy, M. (2001). Efficient variants of the ICP algorithm. In *Proceedings of the International Conference on 3D Digital Imaging and Modeling (3DIM)*, pages 145–152, Quebec City, Quebec, Canada.

Särkkä, S. (2011). Linear operators and stochastic partial differential equations in Gaussian process regression. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, volume 6792, pages 151–158, Espoo, Finland.

Särkkä, S. and Hartikainen, J. (2012). Infinite-dimensional kalman filtering approach to spatio-temporal Gaussian process regression. In *Journal of Machine Learning Research Workshop and Conference Proceedings*, volume 22, pages 993–1001, La Palma, Canary Islands.

Sastry, S. (1999). *Nonlinear Systems: Analysis, Stability, and Control*. Springer, New York, NY, USA.

Se, S., Ng, H.-K., Jasiobedzki, P., and Moyung, T.-J. (2004). Vision based modeling and localization for planetary exploration rovers. In *Proceedings of the 55th International Astronautical Congress*, Vancouver, BC, Canada.

Sherman, J. and Morrison, W. J. (1950). Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *Annals of Mathematics and Statistics*, 21(1):124–127.

Shoemake, K. (1985). Animating rotation with quaternion curves. *ACM SIGGRAPH Computer Graphics*, 19(3):245–254.

Shuster, M. (1993). A survey of attitude representations. *Journal of the Astronautical Sciences*, 41(4):439–517.

Sibley, G., Matthies, L., and Sukhatme, G. (2010a). Sliding window filter with applications to planetary landing. *Journal of Field Robotics*, 27(5):587–608.

Sibley, G., Mei, C., Reid, I., and Newman, P. (2010b). Vast-scale outdoor navigation using adaptive relative bundle adjustment. *International Journal of Robotics Research*, 29(8):958–980.

Simon, D. (2006). *Optimal State Estimation: Kalman, H_∞, and Nonlinear Approaches.* John Wiley & Sons, Inc., New York.

Smith, R., Self, M., and Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles*, pages 167–193.

Snelson, E., Rasmussen, C. E., and Ghahramani, Z. (2004). Warped Gaussian processes. In *Advances in Neural Information Processing Systems (NIPS).*

Solak, E., Murray-Smith, R., Leithead, W. E., Leith, D. J., and Rasmussen, C. E. (2003). Derivative observations in Gaussian process models of dynamical systems. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1033–1040.

Stachniss, C., Plagemann, C., and Lilienthal, A. (2009). Learning gas distribution models using sparse Gaussian process mixtures. *Autonomous Robots*, 26(2-3).

Stillwell, J. (2008). *Naive Lie Theory.* Springer.

Strasdat, H., Montiel, J. M. M., and Davison, A. J. (2012). Visual SLAM: Why filter? *Image and Vision Computing.*

Stuelpnagel, J. (1964). On the parameterization of the three-dimensional rotation group. *SIAM Review*, 6(4):422–430.

Sun, Y., Paik, J., Koschan, A., Page, D., and Abidi, M. (2003). Point fingerprint: A new 3-D object representation scheme. *IEEE Transactions On Systems, Man, And Cybernetics B*, 33(4):712–717.

Taati, B., Bondy, M., Jasiobedzki, P., and Greenspan, M. (2007). Variable dimensional local shape descriptors for object recognition in range data. In *Proceedings of the 11th IEEE International Conference on Computer Vision*, pages 1–8, Rio de Janeiro, Brazil.

Tera, F., Papanastassiou, D. A., and Wasserburg, G. J. (1974). Isotopic evidence for a terminal lunar cataclysm. *Earth and Planetary Science Letters*, 22(1):1–21.

Thrun, S., Fox, D., Burgard, W., and Dellaert, F. (2001). Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141.

Thrun, S. and Montemerlo, M. (2006). The GraphSLAM algorithm with applications to large-scale mapping of urban structures. *International Journal of Robotics Research (IJRR)*, 25(5-6):403–429.

Tong, C. and Barfoot, T. D. (2011a). Batch heterogeneous outlier rejection for feature-poor SLAM. In *Proceedings of the IEEE Conference on Robotics and Automation (ICRA)*, pages 2630–2637, Shanghai, China.

Tong, C. and Barfoot, T. D. (2011b). A self-calibrating ground-truth localization system using retroreflective landmarks. In *Proceedings of the IEEE Conference on Robotics and Automation (ICRA)*, pages 3601–3606, Shanghai, China.

Tong, C. and Barfoot, T. D. (2013a). Evaluation of heterogenous measurement outlier rejection schemes for robotic planetary worksite mapping. *Acta Astronautica*, 88:146–162.

Tong, C. and Barfoot, T. D. (2013b). Gaussian Process Gauss-Newton for 3D laser-based visual odometry. In *Proceedings of the IEEE Conference on Robotics and Automation (ICRA)*, pages 5184–5191, Karlsruhe, Germany.

Tong, C., Barfoot, T. D., and Dupuis, E. (2011). 3D SLAM for planetary worksite mapping. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Invited Symposium on "Field Robotics II"*, pages 631–638, San Francisco, CA, USA.

Tong, C., Barfoot, T. D., and Dupuis, E. (2012a). Three-dimensional SLAM for mapping planetary work site environments. *Journal of Field Robotics, Special issue on "Space Robotics"*, 29(3):381–412.

Tong, C., Dong, H., and Barfoot, T. D. (2013a). Pose interpolation for laser-based visual odometry. Submitted to the *Journal of Field Robotics*.

Tong, C., Furgale, P., and Barfoot, T. D. (2012b). Gaussian Process Gauss-Newton: Non-parametric state estimation. In *Proceedings of the 9th Conference on Computer and Robot Vision (CRV)*, pages 206–213, Toronto, Ontario, Canada.

Tong, C., Furgale, P., and Barfoot, T. D. (2013b). Gaussian Process Gauss-Newton for non-parametric simultaneous localization and mapping. *International Journal of Robotics Research (IJRR)*, 32(5):507–525.

Tong, C., Gingras, D., Larose, K., Barfoot, T. D., and Dupuis, E. (2013c). The Canadian planetary emulation terrain 3D mapping dataset. *International Journal of Robotics Research (IJRR)*, 32(4):389–395.

Vasudevan, S., Ramos, F. T., Nettleton, E., and Durrant-Whyte, H. (2009). Gaussian process modeling of large-scale terrain. *Journal of Field Robotics*, 26(10):812–840.

Wang, J. M., Fleet, D. J., and Hertzmann, A. (2008). Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(2):283–298.

Weisbin, C., Moeller, R., Zimmerman, W., Smythe, W., Van Houten, T., Lincoln, W., Smith, J., Elfes, A., and Adumitroaie, V. (2007). Analysis of concepts for large-scale robotic lunar precursor missions. In *Proceedings of the IEEE International Conference on System of Systems Engineering*, pages 1–6, San Antonio, Texas, USA.

Wettergreen, D., Jonak, D., Kohanbash, D., Moreland, S., Spiker, S., and Teza, J. (2009). Field experiments in mobility and navigation with a lunar rover prototype. In *Proceedings of the 7th International Conference of Field and Service Robotics*, pages 489–498, Cambridge, Massachusetts, USA.

Williams, J. and Bennamoun, M. (2001). Simultaneous registration of multiple corresponding point sets. *Computer Vision and Image Understanding*, 81(1):117–142.

Woodbury, M. (1950). Inverting modified matrices. Memorandum Report 42, Statistical Research Group, Princeton University.

Wulf, O., Nüchter, A., Hertzberg, J., and Wagner, B. (2008). Benchmarking urban six-degree-of-freedom simultaneous localization and mapping. *Journal of Field Robotics*, 25(3):148–163.

Zhang, Z. (1997). Parameter estimation techniques: A tutorial with application to conic fitting. *Image and Vision Computing Journal*, 15(1):59–76.

Zlot, R. and Bosse, M. (2012). Efficient large-scale 3D mobile mapping and surface reconstruction of an underground mine. In *Proceedings of the International Conference on Field and Service Robotics (FSR)*, Matsushima, Japan.

Zuiderveld, K. (1994). Contrast limited adaptive histogram equalization. In Heckbert, P., editor, *Graphics Gems IV*, pages 474–485. Academic Press Professional, San Diego, CA, USA.