UNIVERSITY OF TECHNOLOGY, SYDNEY

# Large-Scale Continuous 2.5D Robotic Mapping

by

Liye Sun

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

at the
Centre for Autonomous Systems
Faculty of Engineering and Information Technology
**University of Technology Sydney**

February 2018

# Declaration of Authorship

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signed: Production Note:
Signature removed prior to publication.

Date: 22/02/2018

# Large-Scale Continuous 2.5D Robotic Mapping

by

Liye Sun

Submitted to the Faculty of Engineering and Information Technology
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

## Abstract

Autonomous robotic systems require building representations of the environment in order to accomplish their particular tasks. Creating rich, continuous probabilistic maps is essential for the robot to perceive the world. As the complexity of the task increases, robots need more sensors or more data to build maps. Noisy and incomplete data is common for the robotic sensors outputs; Gaussian Process (GP), a flexible and powerful statistical model, has become a popular method to cope with the incompleteness of sensory information, incorporate and handle uncertainties appropriately and allow a multi-resolution representation of space. GP regression has been applied in robotic mapping to predict spatial correlations and fill in gaps in unknown areas across the field. The key component of GP for robotic mapping is that it captures spatial correlations and thus increases the accuracy of the representation when fusing data. When multiple sources of data are available, spatial correlations also can be used in fusion to improve accuracy. For large datasets, however, exploiting correlations can become prohibitively expensive. One attractive strategy for reducing storage and computational cost is submapping, which works by dividing the environment into small regions. If no information is shared between maps, submaps are statistically independent.

This thesis investigates how to effectively and efficiently model the necessary spatial correlations that are required to build accurate large-scale maps. Three near-optimal probabilistic mapping frameworks that exploit global and local strategies such as submapping are proposed. **SubGPBF**

applies submapping techniques imposing the conditional independence between submaps. It develops a novel approach to propagate information forward and backwards, which allows spatial correlations to be transferred between submaps after fusing sensor data only within submaps. **GMRF-BF** is a global mapping approach, which exploits the inherent structure of the recently proposed continuous Gaussian Markov Random Field (GMRF) and Bayesian fusion in information form, to model spatial correlations using a sparse information matrix. This leads to information-form Bayesian fusion that is linear in cost. To further increase computational efficiency, this thesis combines ideas from the previous two approaches (the GMRF model and the information-form submapping) to propose another new framework named **subGMRF-BF**. The forward and backward update algorithms formulated in information form are introduced to produce a highly efficient approach due to the conditional independence between submaps when assuming Gaussian distribution.

All three frameworks lend themselves to generate accurate 2.5D probabilistic maps at high resolution. They can handle varying noise from disparate sensor sources and incorporate spatial correlations in a statistically sound way. They are all efficient in memory requirements as there is no need to recover the full covariance matrix or information matrix.

The performance of the three frameworks was evaluated on one controlled terrain elevation dataset and a real water pipe thickness dataset. Five other methods, including one optimal global (fully correlated) method and one without spatial correlations, were used to benchmark the proposed methods. The experiments show that the accuracy, reliability and consistency are improved when the spatial correlations are correctly modelled and incorporated. The experiments also show that all three frameworks achieve storage and computational gain compared with the fully correlated benchmark approach, while subGMRF-BF outperforms all others.

# Acknowledgements

First and foremost I would like to express my sincere gratitude to my supervisors, Prof. Jaime Valls Miro and Dr Teresa Vidal-Calleja, for being enthusiastic in guiding my research, for the freedom to learn and explore exciting topics, and for their patience. I hope I continue to collaborate with you in the future.

I own many thanks to our research centre - Centre for Autonomous Systems (CAS). CAS is a creative, cooperative, smart research centre and also a warm family to me. I met so many knowledgeable people and made so many good friends in CAS, and they have accompanied me through the past three-and-a-half years. It has been great living in beautiful Sydney and spending time with all of them.

Special thanks to Kasra Khosoussi, Maani Ghaffari Jadidi, Teng Zhang and Linh Van Nguyen for the numerous and helpful discussions. Thanks to my friend and colleague Lei Shi, who acted like a big brother to me and guided me through a lot in both study and life. Thanks to my dear friend Raphael Falque, who has been a close friend and been involved in the same industrial project with me for three years. Thanks to Lakshitha Dantanarayana a.k.a. "Lakipedia", for his friendship and vast knowledge on nearly everything I wonder about.

In addition, many thanks to my friends and labmates Liang Zhao, Wenjie Lu, Kanzhi Wu, Daobilige Su, Buddhi Wijerathna, Nalika Ulapane, Andrew To, Mahdi Hassan, Phillip Quin, Antony Tran, Alen Alempijevic, Bradley Skinner, Alexander Virgona, Dinuka Abeywardena, Lasitha Piyathilaka, James Poon, Cédric Le Gentil, Karthick Thiyagarajan, and many others.

I thank my parents and my husband, Kanzhi, for their unconditional love and care. I would not have made it this far without them. It is also a unique and great experience that Kanzhi and I had been sitting in seats face-to-face or back-to-back with each other in CAS lab for two years. We support each other in both life and study. The only thing I regret is that we never had a chance to jointly work on a publication that has both of our names on it.

# Contents

# List of Figures

# List of Tables

# List of Algorithms

# Acronyms

| | |
|---|---|
| 1D | One-Dimensional. xvii, 112 |
| 2.5D | Two-and-a-half Dimensional. xvii, 1, 13–16, 18, 25, 38, 50, 53, 75–78, 86, 107, 128 |
| 2D | Two-Dimensional. xvii, 2, 15, 16, 24, 50, 54, 74, 75 |
| 3D | Three-Dimensional. xvii, 1, 16, 22, 25, 74, 89, 112 |
| ARD | Automatic Relevance Determination. xvii |
| ASV | Autonomous Surface Vehicle. xvii |
| AUC | the Area Under the receiver operating characteristic Curve. xvii |
| BCM | Bayesian Committee Machine. xvii, 6, 21, 22, 26, 87, 101 |
| CAS | Centre for Autonomous Systems. vii, xvii |
| CCIS | Covariance-form Conditionally Independent Submapping. xvii, 8 |
| CDED | Canadian Digital Elevation Data. xvii, 89 |
| CI | Conditionally Independent. xvii, 4–8, 25, 26, 34, 53, 55, 58, 59, 62–64, 66, 68, 69, 77, 78, 80, 81, 83, 84, 86, 96, 105, 107, 108 |

| | |
|---|---|
| CI property | Conditional Independence property. xvii, 4–6, 25, 30, 44, 58, 59, 61, 64, 69, 77, 78, 82, 83, 85, 86, 108–110, 117, 123, 131, 137, 145, 146 |
| CML | Concurrent Mapping and Localisation. xvii |
| COM | Continuous Occupancy Map. xvii |
| DAG | Directed Acyclic Graph. xvii, 34 |
| DEM | Digital Elevation Model. xvii |
| DGM | Directed Graphical Model. xvii, 34, 55 |
| EK | Expected Kernel. xvii |
| EKF | Extended Kalman Filter. xvii, 12, 13, 48, 66 |
| ESM | Expected Sub-Map. xvii |
| GM | Graphical Model. xvii, 34, 44, 55 |
| GMRF | Gaussian Markov Random Field. vi, x, xvii, 4–8, 15–17, 24, 25, 27, 43–47, 52, 72–74, 76, 77, 80–82, 86, 93, 94, 101, 105, 108–110 |
| GMRF-BF | Gaussian Markov Random Fields for Bayesian Fusion. xvii, 3–6, 8, 25, 26, 71, 72, 75–78, 88, 96, 99, 101–103, 105, 106, 108–111 |
| GMRF-SPDE | Gaussian Markov Random Field via the Stochastic Partial Differential Equation. xvii, 17, 43, 45, 46, 71–73, 91, 93, 109, 110 |
| GP | Gaussian Process. v, x, xvii, 2–7, 15, 16, 18–22, 24–27, 29, 36–43, 50–55, 59, 60, 63, 68, 69, 71, 73, 76, 88, 91, 93, 101, 105, 107, 109–112, 143, 144 |
| GPBF | Gaussian Processes for Bayesian Fusion. xvii, 7, 24–26, 29, 50–52, 72, 76, 77, 87, 88, 96, 101–103, 105, 107, 111, 112 |
| GPIS | Gaussian Process Implicit Surfaces. xvii, 112 |

| | |
|---|---|
| GPOM | Gaussian Processes Occupancy Map. xvii, 16 |
| GRF | Gaussian Random Field. xvii, 16, 17, 43, 45, 46, 71, 127 |
| GT | Groundtruth. xvii, 90, 91 |
| HMM | Hidden Markov Model. xvii |
| i-backwardUpdate | Information-form Backward Update. xvii |
| i-correlateFusion | Information-form correlated Bayesian fusion. xvii |
| i-forwardUpdate | Information-form Forward Update. xvii |
| I-GPOM | Incremental Gaussian Processes Occupancy Map. xvii |
| ICIS | Information-form Conditionally Independent Submapping. xvii, 8 |
| iff | if and only if. xvii, 33, 34, 38, 44 |
| iid | independent and identically distributed. xvii, 38, 67, 90 |
| INLA | Integrated Nested Laplace Approximation. xvii, 45 |
| KLD | Kullback-Leibler Divergence. xvii |
| MAP | Maximum a Posteriori. xvii, 12, 48, 51, 59, 72, 74 |
| MCMC | Markov chain Monte Carlo. xvii, 45 |
| MDP | Markov Decision Process. xvii |
| MRF | Markov Random Field. xvii, 16, 17, 24 |
| MSE | Mean Squared Error. xvii, 39 |
| MVN | Multivariate normal. xvii, 4–6, 32, 37, 38, 44, 45, 52, 55, 60, 61, 67, 82, 101, 108, 110, 111, 125 |

| | |
|---|---|
| NDT | Non Destructive Testing. xvii |
| NLML | Negative Log of the Marginal Likelihood. xvii |
| OGM | Occupancy Grid Map. xvii |
| PD | Positive Definite. xvii |
| pdf | Probability Density Function. xvii, 30, 32, 44, 54, 60, 61, 81, 84, 123, 144 |
| POMDP | Partially Observable Markov Decision Process. xvii |
| PSD | Positive Semi-Definite. xvii |
| RMSE | Root Mean Squared Error. xvii, 105 |
| ROC | Receiver Operating Characteristic. xvii |
| ROS | Robot Operating System. xvii |
| SE | Squared Exponential. xvii, 41, 42 |
| SEIF | Sparse Extended Information Filter. xvii, 4 |
| SLAM | Simultaneous Localisation And Mapping. xvii, 4, 6, 11–13, 25, 27, 48, 58, 66 |
| SPDE | Stochastic Partial Differential Equation. xvii, 17, 45, 47, 76, 81, 127 |
| subGMRF-BF | Gaussian Markov Random Fields for Bayesian Fusion with Conditionally Independent 2.5D Submapping. xvii, 3, 6, 8, 25–27, 75, 78, 86, 96, 101, 102, 105, 106, 108–112 |
| subGPBF | Gaussian Processes for Bayesian Fusion with Conditionally Independent 2.5D Submapping. xvii, 3, 6–8, 25, 53–55, 65–69, 73, 75–78, 86, 96, 101, 102, 105, 106, 108–112 |

| | |
|---|---|
| UGM | Undirected Graphical Model. xvii, 55 |
| w.r.t. | with respect to. xvii, 40, 44 |
| WGP | Warped Gaussian Process. xvii |
| WGPOM | Warped Gaussian Processes Occupancy Map. xvii |

# Nomenclature

| | |
|---|---|
| $\approx$ | approximately equal |
| $\mathcal{O}(\cdot)$ | big O notation, which characterizes the growth rates of functions; for functions $f$ and $g$ on $\mathbb{R}$, we write $f(n) = \mathcal{O}(g(n))$ if the ratio $f(n)/g(n)$ remains bounded as $n \to \infty$ |
| $\boldsymbol{\theta}$ | vector of hyperparameters |
| $\boldsymbol{\mu}_b^a$ | the mean estimate $\boldsymbol{\mu}$ of variable $b$ given observation $a$ |
| $\boldsymbol{\xi}_{s_1}$ | state vector variables of submap $s_1$ |
| $\boldsymbol{x} \sim p(\boldsymbol{x})$ | $\boldsymbol{x}$ is distributed according to distribution $p(\boldsymbol{x})$ |
| $\boldsymbol{y}_{-ij}$ | the set of variables $\boldsymbol{y}$ = except $y_i$, $y_j$ |
| $\perp$ | independent |
| $\mathbb{Cov}[\cdot]$ | covariance of random variables/vectors |
| $\mathbb{E}[\cdot]$ | expected value of a random variable |
| $\iff$ | if and only if |
| $\mathbb{R}$ | the real numbers |
| $\mathbf{x}$ | a column vector |
| $\mathcal{N}_c(\boldsymbol{\eta}, Q)$ | equals to $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$, where $Q = \Sigma^{-1}$, $\eta = Q\boldsymbol{\mu}$ |
| $\mathrm{tr}(A)$ | trace of matrix $A$ |

| | |
|---|---|
| $\propto$ | proportional to |
| $\sim$ | distributed according to |
| $\top$ | transpose operation |
| $\triangleq$ | an equality which acts as a definition |
| $\mathbb{V}[\cdot]$ | variance of a random variable |
| $\{x_i\}_{i=1}^n$ | a set that equals to $\{x_i, \ldots, x_n\}$ |
| $I_n$ | an identity matrix of dimension $n \times n$ |
| $k(\cdot, \cdot)$ | covariance function of Gaussian processes |
| $s_1^+$ | the updated estimate of $s_1^-$ |
| $s_1^-$ | the prior estimate of submap $s_1$ |
| $s_1^{++}$ | the updated estimate of $s_1^+$ |
| $X$ | a matrix, or a matrix of training inputs |
| $x$ | a scalar |
| $X^*$ | a matrix of test inputs |
| $X^\top$ | the transpose of matrix $X$ |
| $X^{-1}$ | the inverse of matrix $X$ |
| $x_i$ | the $i$-th element of $\mathbf{x}$ |
| iff | if and only if |

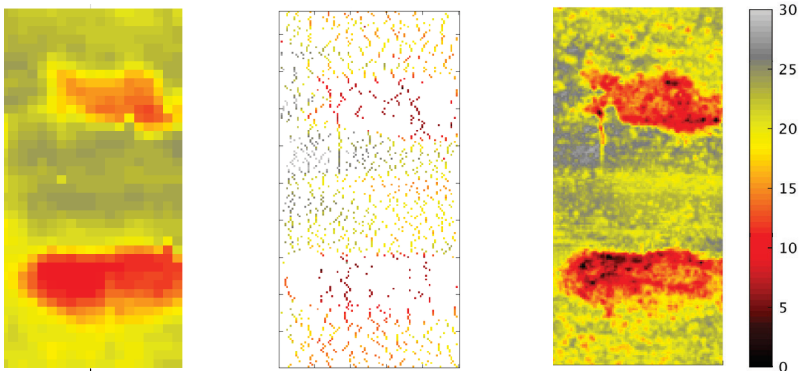
# Chapter 1

# Introduction

## 1.1 Motivation

Constructing accurate maps of environments remains a fundamental yet challenging task for mobile robots. The mapping problem is arguably regarded as one of the most significant issues in the pursuit of attaining truly autonomous mobile robots. The generation of reliable and accurate spatial models of the robot's surroundings is central to the goals of localisation (Borenstein et al. 1996), manipulation (Kortenkamp et al. 1998) and path planning (Elfes 1989). There are various kinds of representations for different tasks. For example, some commonly used Two-and-a-half Dimensional (2.5D) representations include height maps (Gutmann et al. 2005), multi-level surface maps (Triebel et al. 2006), occupancy maps (Elfes 1989) and pipe wall thickness maps (see Figure 1.1). Since robots live in the Three-Dimensional (3D) world, in recent years, 3D representations, such as voxel grid maps (Wurm et al. 2010), point clouds (Rusu et al. 2008) and meshes (Newcombe and Davison 2010), are getting more popular. Different representations, generated with various sensors or approaches, can for instance be at different resolutions, e.g. Figure 1.1, and they can be combined to improve the accuracy and reduce the uncertainty (Mahler 2007; Durrant-Whyte and Henderson 2008).

A key challenge in robotic mapping arises from how to model the *spatial correlation*. The spatial correlation represents the relationship between nearby spatial locations. Tobler's first law




(A) (B) (C)

FIGURE 1.1: Two-Dimensional (2D) view of pipe wall thickness maps. (A) Complete measurements located at uniform grids. (B) Incomplete measurements located at non-uniform grids. (C) Complete measurements located at uniform, finer grids.

of geography states that "everything is related to everything else, but near things are more related than distant things" (Tobler 1970, p. 236). One of the main reasons why spatial correlations are necessary is that robotic mapping problems usually assume the observed data to be statistically independent to make the problem relatively easy to solve. Unfortunately, in the real world, correlations exist and the variables being mapped are statistically dependent. For example, in Figure 1.1a, the small thickness values are close in 2D locations to each other (see the red areas). Accounting for the correlations properly during mapping is the key to building maps successfully.

Incorporating spatial correlations in maps comes with the issue of efficiency, especially when mapping large environments or using all the information from high-resolution sensors. For example, volumetric laser scanners or depth cameras may generate millions of points in a single scan; even if correlations are not taken into account, these sensors require efficient mapping algorithms. However, considering the spatial correlations among all data points incurs a heavy computation burden. How to overcome this issue remains an open problem.

In recent years, Gaussian Process (GP) (Rasmussen and Williams 2006), one of the state-of-the-art machine learning techniques for regression and classification, has been applied to probabilistic robotic mapping, e.g. terrain mapping (Vasudevan et al. 2009), occupancy mapping (O'Callaghan and Ramos 2012) and surface reconstruction (Kim and Kim 2014). Compared to the traditional deterministic approaches for mapping, the GP based methods open up the door to stochastic

and data-driven approaches. In mapping applications, GP is applied to learn the noise-free latent process together with the spatial correlations from the noisy input data. The prediction at unknown spatial locations is treated as a regression problem and takes into account the correlations, which are measured via the covariance. Intuitively, known points close to a query point should contain information about the query point's value. The GP predicted probabilistic maps can be further used for grasping (Mahler et al. 2015), path planning and exploration (Jadidi et al. 2013). Meanwhile, Vasudevan (2012) has applied GP for data fusion by regarding the data from multiple sources as different samples of a common underlying terrain. Later, Vidal-Calleja et al. (2013) propose Gaussian Processes for Bayesian Fusion (GPBF), in which spatial correlations are learned through a GP to generate a prior map which subsequently will be used in a Bayesian fusion framework for mapping with multiple sources of information.

Particularly, a GP models the full spatial correlations of all data, which can be thought of as an entirely connected Bayes network (Koller and Friedman 2009, Chapter 3); therefore, a wide-range of algorithms can be applied to obtain accurate and reliable robotic maps. However, one of the major disadvantages of GP based approaches is the high memory requirements and computational complexity due to the allocation and inversion of the covariance matrix. For instance, GP for fusion in Vasudevan (2012) requires inverting the dense covariance matrix of all data in each iteration of the hyperparameter optimisation. This gives the $\mathcal{O}(N_{all}^3)$ time and $\mathcal{O}(N_{all}^2)$ memory complexity per iteration, where $N_{all}$ is the total number of all sources of data. Therefore, the GP based approaches are not directly applicable to large-scale robotic mapping.

## 1.2 Research Problems and Scope

To reduce the computational complexity of grid mapping using GP, this thesis develops efficient algorithms which maintain the necessary spatial correlations to model the underlying process of observed data. The research objectives also include taking into account the noisy, sparse and large volumes of data; and building continuous high-resolution[1] 2.5D grid maps. Three methods, named Gaussian Processes for Bayesian Fusion with Conditionally Independent 2.5D Submapping (subGPBF), Gaussian Markov Random Fields for Bayesian Fusion (GMRF-BF) and Gaussian Markov Random Fields for Bayesian Fusion with Conditionally Independent 2.5D Submapping

[1] *A high-resolution map* is a map that is sharp and finely detailed rather than blurry and inexact, e.g. Figure 1.1c.

(subGMRF-BF), are developed for building large-scale 2.5D grid maps via data fusion. GMRF-BF is a global method and the other two are based on submaps. The three of them are different while having some underlying connections.

The mathematical basis for this thesis is primarily based on the Bayesian approaches, the properties of Multivariate normal (MVN) distributions and the Conditional Independence property (CI property). Characteristics of these mathematical and statistical tools are as follows:

- ◇ Using Bayesian approaches allows for the realistic and accurate estimation of the uncertainty, and it is easy to incorporate the prior information into the current estimation. The Bayesian approaches used here include, but are not limited to, Bayes rule, which has long been used in data fusion, and GP. Despite the growing prevalence of Bayesian fusion, it has rarely been studied in order to gain memory and computational efficiency when involving the spatial correlation within Bayesian fusion for grid mapping.

- ◇ MVN distributions are widely used mainly due to their succinct analytical properties, which often come with a closed-form solution. In Simultaneous Localisation And Mapping (SLAM), the research concerning information-form MVN distributions[2] has given rise to an important estimator - Sparse Extended Information Filter (SEIF) (Thrun et al. 2004), which allows for efficient, scalable SLAM. The key is to exploit the sparsity in the information form as opposed to using the dense covariance matrix. This sparse information matrix is reminiscent of a Gaussian Markov Random Field (GMRF). However, due to the complexity of defining and solving the continuous GMRF model, to the best of our knowledge, GMRF has not been applied for continuous robotics mapping.

- ◇ Conditional Independence (CI) can be exploited in some different ways, e.g. it is characteristic of the SLAM problem and in turn leads to a factored representation (Montemerlo et al. 2002; Walter et al. 2007). Piniés and Tardós (2008) impose the CI property between submaps, i.e. decomposed global map, for large-scale SLAM. However, their work was developed in a SLAM context. This thesis only focuses on the Mapping problem, where transition and measurement models are not available and correlations have to be learned through a machine learning technique, i.e. GP. It is worth exploring how to generalise the Conditionally Independent (CI) submapping technique to such cases.

[2]This is sometimes referred to as the canonical form for multivariate Gaussian distributions.

It is worth noting that in this thesis it is assumed that the locations of sensor measurements are known and accurate. It is also assumed that the variables are MVN distributed and the sensor observation model is linear Gaussian. Besides, the CI property of submaps is imposed.

## 1.3 Main Contributions

The main contribution of this thesis is to allow the use of correlations for large-scale discrete and continuous mapping problems through the design and development of efficient approaches that exploit either the CI property, the information form or both for MVN distributions. The specific contributions are as follows:

- **SubGPBF**
  - Development of a unified framework for large-scale 2.5D mapping, which can handle arbitrarily large maps. It side-steps the high computational and memory cost induced by incorporating the spatial correlation within GP interpolation and Bayesian fusion via building CI submaps.
  - Design of a new pair of correlation propagation algorithms, which not only enables information to be transmitted between submaps bidirectionally, but also can recover the optimal global map given the CI condition. In addition, each submap is ensured to be always currently optimal.
  - SubGPBF is easy to use and generally applicable, and two variants: probabilistic prediction and incremental online mapping are described in the thesis.
- **GMRF-BF**
  - Development of a unified framework, which studies the effect of the information-form MVN variables on substantially reducing the computation complexity. The GMRF modelling and the information-form Bayesian fusion are seamlessly used together in an algorithm that only requires square time complexity.
  - Introduction of the continuously indexed GMRF approach, which originated from the spatial statistics field, into robotic mappings and the use of it for modelling the spatially correlated, noisy sensor data.

  - Investigation of the explicit link between the hyperparameters of GP and the continuously indexed GMRF.

◇ **SubGMRF-BF**

  - Development of a unified framework which has gained further computational advantages by imposing the CI property between the information-form MVN submaps. This is inspired by the fact that the CI property and spatial correlations are encoded in the sparse information matrix, whose sparsity enables efficient computations.
  - Investigation and the use of the close links of the Bayes network, the CI property, the sparsity structure of the information matrix in the GMRF representation.
  - Design of a novel pair of information-form correlation propagation algorithms, which not only enables information to be transmitted between submaps bidirectionally, but also can recover the optimal global map, given the CI condition, in constant time. Submaps are always currently optimal.

SubGPBF and subGMRF-BF have been developed based on local strategies while GMRF-BF is a global approach. GMRF-BF and subGMRF-BF exploit the computational efficiency in the information form. SubGMRF-BF achieves improved efficiency in building submaps when compared with subGPBF. In short, all three methods address main limitations of GPs particularly in terms of scalability to large sets of query points.

## 1.4 Thesis Structure

Figure 1.2 shows an overview of the structure of this thesis. The detailed outline of each chapter follows:

**Chapter 2** reviews the related work in probabilistic robotic mapping and statistical models. Firstly, SLAM and some of concepts used in this thesis are briefly introduced, followed by the discussion of two Bayesian approaches, GP and GMRF, for modelling the imperfect data and mapping. Then, probabilistic data fusion approaches, including those with GP and Bayesian

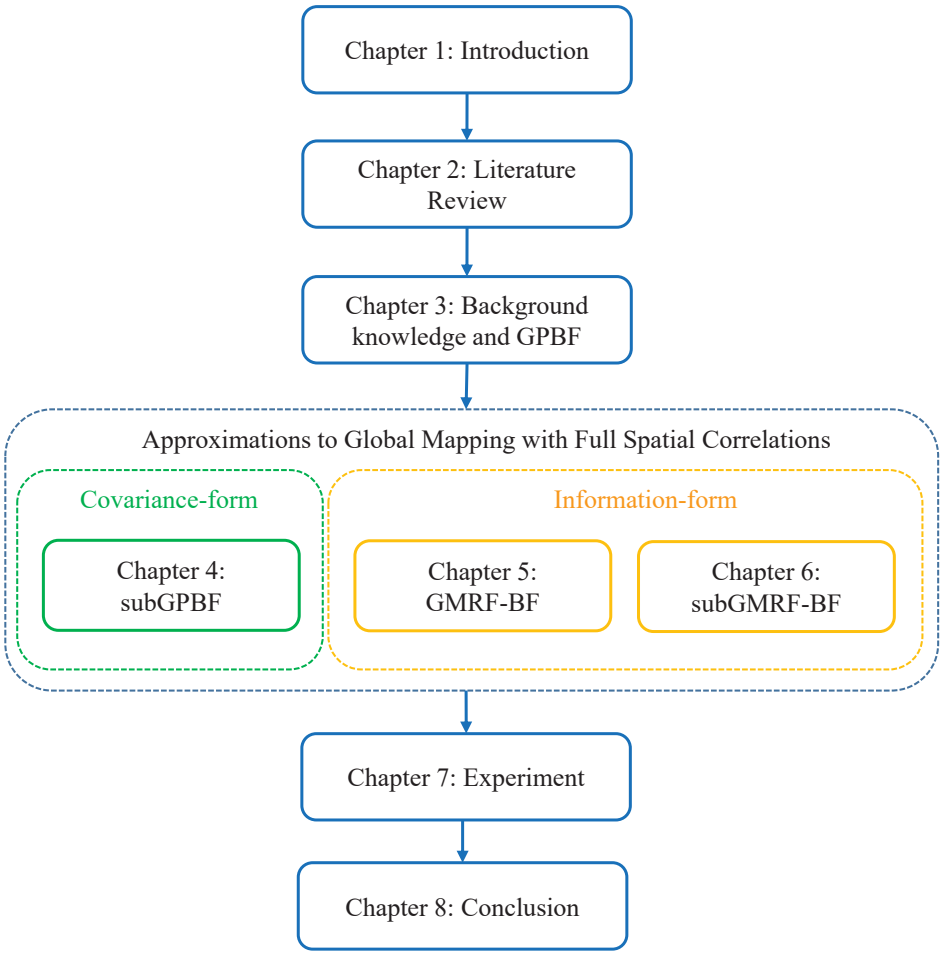



FIGURE 1.2: Overall thesis structure.

Committee Machine (BCM), are discussed. Next, some approximation methods which address the scalability problem are discussed. The link between some literature work and this thesis is then explained.

**Chapter 3** describes the mathematical background knowledge used throughout this thesis. The mathematical notations, some basic probability theories, GP, GMRF and Bayesian fusion are briefly introduced. In particular, Section 3.6 formally introduces the Gaussian Processes for Bayesian Fusion (GPBF), which forms the basis and benchmark of our approach.

**Chapter 4** describes subGPBF. Firstly, GP regression for creating prior submaps and Bayesian fusion for updating the submaps are explained. Then the novel correlation propagation algorithms for the CI submapping are elaborated. In addition, two variants of subGPBF, one for prediction and the other for sequential mapping, are shown in Section 4.4.

**Chapter 5** elaborates GMRF-BF, which uses the continuously indexed GMRF for prior mapping, and uses the information-form Bayesian fusion with spatial correlations for updating the prior map. Then, a discussion about the efficient techniques for mean and variance recovery is presented.

**Chapter 6** explains subGMRF-BF, which can be seen as an adaptation of subGPBF in information form or combining GMRF-BF with CI submapping. The focus is on the novel information-form correlation propagation algorithms.

**Chapter 7** evaluates the above three approaches against five other methods on two different datasets. The created maps are presented, and the computational complexity, the real computational time and the map accuracy are compared.

**Chapter 8** summarises the research work, discusses the limitations, and draw the future lines of the work.

## 1.5 Publications

The initial idea of performing CI submapping with Gaussian Processes and Bayesian fusion, referred to as Covariance-form Conditionally Independent Submapping (CCIS), was firstly presented in Sun et al. (2015). However, CCIS cannot guarantee the optimal global map due to the assumption that the last submap is globally optimal. To get the optimal global map, this thesis designs the forward update algorithm and uses it together with CCIS, thus getting subGPBF.

The development of GMRF-BF was presented in (Sun et al. 2016).

The information-form CI submapping with GMRF and Bayesian fusion, named as Information-form Conditionally Independent Submapping (ICIS), was initially presented in (Sun et al. 2017). However, ICIS cannot guarantee the optimal global map due to the assumption that the last submap is globally optimal. Again in this thesis, subGMRF-BF creates the optimal global map by designing a new forward update algorithm and using it together with ICIS.

The publication list in reverse chronological order is as follows:

- **ICRA'17**: Sun, L., Vidal-Calleja, T., and Valls Miro, J. (2017). Coupling Conditionally Independent submaps for large-scale 2.5D mapping with Gaussian Markov Random Fields. In 2017 IEEE International Conference on Robotics and Automation, pages 3131-3137. (Sun et al. 2017)

- **IROS'16**: Shi, L., Valls Miro, J., Zhang, T., Vidal-Calleja, T., Sun, L., and Dissanayake, G. (2016). Constrained sampling of 2.5D probabilistic maps for augmented inference. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3131-3136. (Shi et al. 2016)

- **ICRA'16**: Sun, L., Vidal-Calleja, T., and Valls Miro, J. (2016). Gaussian Markov random fields for fusion in information form. In 2016 IEEE International Conference on Robotics and Automation, pages 1840-1845. (Sun et al. 2016)

- **ICRA'15**: Sun, L., Vidal-Calleja, T., and Valls Miro, J. (2015). Bayesian fusion using Conditionally Independent submaps for high resolution 2.5D mapping. In 2015 IEEE International Conference on Robotics and Automation, pages 3394-3400. (Sun et al. 2015)

- **ACRA'15**: Shi, L., Sun, L., Vidal-Calleja, T., and Valls Miro, J. (2015). Kernel-specific Gaussian Process for predicting pipe wall thickness maps. In Australasian Conference on Robotics and Automation 2015. AARA. (Shi et al. 2015)

# Chapter 2

# Related Work

HOW to learn to construct rich and reliable maps from sensor observations is a fundamental problem in both robotics and computer vision. An accurate map with uncertainty can aid complex tasks during robot exploration, such as path planning and object grasping. Despite significant progress in this area, it still poses great challenges. In practice, most sensors have different limitations, such as low resolution, measurement noise and a limited field of view, which render most sensory data uncertain and incomplete. Meanwhile, as the scale of environments grows or due to the high resolution of sensors, sensor data could quickly become hard to manage.

## 2.1 Localisation and Mapping

Before revisiting the previous contributions in mapping using Gaussian Processes and Gaussian Markov Random Field, we firstly review the related work in the field of Simultaneous Localisation and Mapping (SLAM) (Durrant-Whyte and Bailey 2006; Bailey and Durrant-Whyte 2006). Mapping is SLAM when the localisation is given. The reminder of this subsection summarises the previous work in solving the SLAM problem, specifically focusing on the sub-mapping algorithms.

The objectives of SLAM problem solving consist of two parts: (1) estimate the state of the robot which is equipped with sensors to perceive the environment and (2) reconstruct the map of the environment which the robot is manoeuvred in. The first Localisation problem and the latter

Mapping problem are solved simultaneously rather than sequentially. Generally speaking, the state in the first sub-problem is described using the 2D/3D pose of the robot and the latter one is to represent the map using estimated places of interests such as landmarks, lines or surfaces. However, in the mapping problem, the localisation problem is no longer of interested, and building higher level understanding of the environment is attached with higher priority. For example, in this thesis, the significant focus is to build a continuous map with uncertainty.

The approaches in solving SLAM problem can be categorised into two groups of methods: (1) filter based approaches (Dissanayake et al. 2001; Julier and Uhlmann 2004; Montemerlo et al. 2002) and (2) optimisation based approaches(Grisetti et al. 2010; Olson et al. 2006; Kaess et al. 2008). Given the motion model and the observation model, filter based approaches process the input motion commands and observation measurements sequentially using state propagation and update to estimate the mean and covariance of the state. By linearising the models, Dissanayake et al. (2001) firstly proposed an Extended Kalman Filter (EKF) based solution towards the SLAM problem and analysed its convergence property under first order approximations. Huang and Dissanayake (2007) also provided an analytical illustration towards the inconsistency issue of the EKF-SLAM solution which is caused by the fact the Jacobians cannot be evaluated at the true states. Huang et al. (2008a) proposed an alternative solution to solve the consistency issue by manipulating the Jacobian matrices. Besides EKF-SLAM and its variants, Unscented Kalman Filter SLAM (Huang et al. 2009) and FastSLAM (Montemerlo et al. 2002) have been proposed to formulate and solve the SLAM problem. In contrast, optimisation based approaches, for both feature based SLAM and pose graph SLAM, can converge to the global minimum under various situations and can avoid the inconsistency issue through iterative optimisation. To solve the SLAM problem as a Maximum a Posteriori (MAP) estimation problem, factor graph (Kaess et al. 2012) has been widely accepted to manage the independence among variables.

For the SLAM problem, a key concern to is reduce the complexity of the original problem. Submapping algorithms reduce the complexity by dividing the whole map into groups of state vector variables (features and/or vehicle poses) that are processed separately. Looked from the factor graph perspective, this is divided into different subgraphs and the overall graph is optimised by alternating local optimisation of each subgraph, with a global refinement. Submaps are often assumed to be statistically independent, and they can be consistently joined using Map Joining algorithm (Tardós et al. 2002) or equivalent Constrained Local Submap Filter (CLSF) (Williams

et al. 2002) with joining cost $\mathcal{O}(n^2)$. Later, the Divide and Conquer SLAM (Paz et al. 2008a) provides a more efficient strategy to join local maps with $\mathcal{O}(n)$ cost in exploration. However, the main limitations of these techniques are their inability to share information between maps and a memory cost of $\mathcal{O}(n^2)$. To address these problems, the CI-Graph SLAM approach (Piniés et al. 2009) has been developed upon the Conditionally Independent (CI) Submaps, which were developed by Piniés and Tardós (2008). The technique in Piniés and Tardós (2008) allows local submaps to share submap components and information in a consistent manner. Additionally, the final map obtained is the same as with the classical EKF SLAM algorithm. CI-Graph SLAM is also efficient in memory requirements since it does not need to recover the full covariance matrix.

There are some submapping techniques that reduce the complexity of EKF SLAM via trading off precision (Leonard and Feder 2000). Some of these techniques combine submaps with a graph structure that represents adjacency relations between maps. For instance, in the Atlas Framework (Bosse et al. 2003), nodes of a graph correspond to submaps, and links between nodes represent the relative locations between adjacent submaps. However, in order to achieve high efficiency, there are no loop constraints imposed to update the graph estimation.

## 2.2 Probabilistic 2.5D Mapping with Statistical Tools

### 2.2.1 Probabilistic Mapping

2.5D grid map is one of the favoured choices in both ground robotic mapping (Qiu et al. 2009) and spatial statistics (Diggle et al. 2003; Banerjee and Fuentes 2012). A typical example is the elevation maps, which "store in each cell of a discrete grid the height of the surface at the corresponding place in the environment" (Pfaff et al. 2007). Some approaches also store the variance or uncertainty of height in each cell (Qiu et al. 2009). The benefits include the simplicity, low computation cost, compactness and the low sensitivity to discretisation errors (Tse et al. 2012). Some major drawbacks are the inability to handle abrupt changes, the dependence on grid size and the issue of scalability in large environments.

Practically, many state-of-the-art algorithms for robotic mapping in the literature are probabilistic (Thrun et al. 2005). They all employ probabilistic models and rely on probabilistic inference

for turning sensor measurements into maps. The popularity of probabilistic techniques stems from the fact that robotic mapping is characterised by uncertainty and sensor noise. These two issues are often ubiquitous in robotics as sensor capabilities are limited. Probabilistic algorithms address these problems by explicitly modelling (different sources of) noise and their effects on the measurements. For example, in terrain mapping through elevation maps, Kelly and Stentz (1997) used the concept of a "scatter matrix" to represent the local geometric uncertainty in a grid. Lacroix et al. (2002) developed stereo-vision based elevation maps and recognised that the main problem was uncertainty management. To address this issue, they proposed a heuristic data fusion algorithm on the basis of the Dempster-Shafer theory (Dempster 1967).

A majority of probabilistic robotic mapping approaches have the strict assumption that all grid cells are independent, while observations are often inclined to spatial correlations. As a result, this strict assumption may lead to incorrect results, e.g. sparse sensor measurements yield discontinuous maps. This thesis argues that spatial correlations should be taken into account in probabilistic data models, even if sensor data were collected discretely. This is because the environment is normally locally correlated, and observations at locations in close spatial proximity often tend to be more similar than observations at locations far apart. However, computing the spatial correlations for large-scale datasets usually incurs high computational and storage complexity. Large-scale mapping quickly becomes infeasible in a field robotics scenario, e.g. a mining or space exploration scenario.

There have been many interpolation strategies reported to acquire continuous maps from sparse sensor measurements. Most of the interpolation methods are borrowed from spatial statistics (Cressie 1992; Stein 1999). The choice of interpolation methods can have severe consequences on accuracy. Kidner et al. (1999) reviewed the grid data interpolation methods and recommended applying higher order polynomial interpolation methods. More recently, Li and Heap (2014) reviewed and categorised a broader range of approaches, some of which can handle sparse data on regularly or irregularly spaced samples. In robotics, Ye and Borenstein (2003) used median filtering to fill in the missing data on an elevation map. However, these interpolation methods have tended to focus on filling the missing data; other complications, such as measurement error, have not been considered.

In short, probabilistic 2.5D maps are popular and widely used. However, they have the main

weaknesses of lacking a statistically direct way of coping with data uncertainty, the inability to appropriately describe spatial correlations and to manage big data. The rest of this section introduces two popular statistical approaches, GP and GMRF models, which give possible solutions to address the issues mentioned above.

### 2.2.2 Gaussian Processes for Probabilistic Mapping

In recent years there has been a growing interest in GPs (Rasmussen and Williams 2006) for probabilistic robotic mapping. GP is a flexible non-parametric Bayesian approach[1] for regression and classification. Not only in the robotics field, "GPs have been known for a long time in the statistics and geostatistics field" (Rasmussen and Williams 2006), where it is known under the name *Kriging* (Stein 1999; Schabenberger and Gotway 2004). In 2.5D mapping, GP can be used to learn from the noisy data (2D spatial location and the interested value being the third dimension) to generate a continuous-domain, compact, and non-parametric representation of the targeted environment. The trained GP model can assign the gaps in sensor data with best linear unbiased estimates that are correlated to neighbouring areas covered by the sensor; therefore sensor limitations and occlusions can be overcome. Also, maps at any required resolution can be predicted.

GP regression was firstly introduced into terrain mapping by Lang et al. (2007) and later by Plagemann et al. (2008a,b, 2009) and Vasudevan et al. (2009). The former four implementations used the equivalence of a stationary squared exponential covariance kernel and Vasudevan et al. (2009) used the neural network, to handle both smooth surfaces and the inherent (characteristic) surface discontinuities. Lang et al. (2007) initialised the kernel matrices evaluated at each point with parameters learnt for the corresponding stationary kernel and then iteratively adapted them to account for local structure and smoothness. Plagemann et al. (2008a,b) introduced "hyper-GP" (an independent stationary kernel GP) to predict the most probable length-scale parameters to suit the local structure. Plagemann et al. (2008b, 2009) modelled the space as an ensemble of a GP in order to reduce computational complexity. The measurement model was assumed to be unimodal zero-mean Gaussian with a constant variance throughout data collection. Kjærgaard et al. (2011) proposed a slightly more flexible measurement model, which allowed two hypotheses of

[1] A Bayesian non-parametric model is a Bayesian model on an infinite-dimensional parameter space. (Orbanz and Teh 2010)

measurement noise variances, one for the ground measurements and the other for the measurements around obstacles, within the same dataset. Similar as before, they assumed a parametric zero-mean Gaussian likelihood distribution.

GP classification can be used to attain binary labelled maps, such as occupancy map. O'Callaghan and Ramos (2012) developed Gaussian Processes Occupancy Map (GPOM), which used GPs to infer occupancy probabilities of cells not directly intersected by sensors by making use of the spatial correlations. The continuity property of GPs overcomes the traditional assumption of independence between cells. GPOM was further investigated for planning and exploration by Jadidi et al. (2014, 2015). The uncertainty map that is predicted by the GP could be used to highlight unexplored regions and optimise a robot's search plan. Kim and Kim (2014) adapted GPOM for 3D cases, and they developed a "divide-and-conquer" strategy to reduce the computational cost of building 3D GPOMs.

Although GPs mapping methods enjoy greater flexibility to capture dependency and build reliable maps, yet they are not suitable for online use (Thrun 2003). The primary cause is that GPs scale cubically in the number of training data and squarely with storage demands when performing exact inference. The memory requirements of GPs will increase by several orders of magnitude when the map resolution or the size of mapped environments increases. For large problems (data number bigger than 10000) both storing the covariance matrix and inverting it are prohibitive on modern workstations.

### 2.2.3 Gaussian Markov Random Fields for Probabilistic Mapping

In Section 2.2.2, GPs mapping methods treat the domain as a continuous Gaussian Random Field (GRF). On discrete domains, such as grids or more generally any collection of countably 2D locations, a popular choice is to use GMRFs.

GMRFs are discrete GRFs equipped with the Markov property or the Markov Random Fields (MRFs) combined with the Gaussian assumption (Rue and Held 2005). GMRFs possess appealing computational properties due to the sparsity of the information matrices. GMRFs have been extensively used in the analysis of 2.5D data in different areas of spatial statistics, yet there has been little research in applying GMRFs for statistically modelling the spatially correlated data

in robotic grid mapping. Nevertheless, there has been some study in robotics that use standard MRFs to express spatial correlations. For instance, Wellington et al. (2005) included multiple MRFs, which interact through a hidden semi-Markov model, to enforce relative smoothness of labellings. The trained model can make inference at unknown places. Their results show that including the correlations significantly improves obstacle classification accuracy. Tse et al. (2012, 2015) designed a MRF model which incorporates sensor uncertainties and enables probabilistic fusion between sensor and terrain information. Local spatial correlations were modelled by the numerically defined clique potential functions of neighbourhoods, while the information matrix was not used at all. All these three examples require manually defining the complex clique potentials before solving the sophisticated graphical model. Such methods are not generally applicable for mapping, and the computation cost of optimisation can be high. Another concern is that none of these methods shows or investigates the connections between MRFs and GRFs when a joint Gaussian distribution is assumed.

While the basic MRFs are restricted to discrete, regular grids, much research work has been carried out to make MRFs continuous in the Gaussian field. For instance, Rue and Tjelmeland (2002) showed that general stationary covariance models could be closely approximated by MRFs by numerically minimising errors in the resulting covariances. A drawback of the methods is that the numerical optimisation must, in general, be performed for each distinct parameter configuration. Recently, Lindgren et al. (2011) has derived a method for building an explicit and continuous Markov representation of a Gaussian Matérn field (a GRF with Matérn covariance function), named as Gaussian Markov Random Field via the Stochastic Partial Differential Equation (GMRF-SPDE). The method uses the fact that Gaussian Matérn field is a solution to a certain Stochastic Partial Differential Equation (SPDE). By considering weak solutions to this SPDE concerning some set of local basis functions, an approximation is obtained. GMRF-SPDE re-formulates the GMRF model in combination with the finite element method from numerical analysis to account for randomly located data, without the limitation to uniform grid maps. Besides, the information matrix of a GMRF model, which is sparse, can be written directly as a function of the parameters, without any need for costly numerical calculations. Besides, Lindgren and Rue (2015) has shown that GMRF-SPDE can cope with the uncertain and incomplete data while modelling the spatial correlations. GMRF-SPDE has been widely applied in different areas of geostatistics, and the most recent survey is in Rue et al. (2017).

## 2.3 Probabilistic Fusion for 2.5D Mapping

To estimate an accurate representation of the environment, it is usually necessary to fuse measurements obtained at different times while a robot moves, or from multiple sensors at the same time. In a robotic system, probabilistic fusion methods are usually needed to combine data, express data uncertainty and handle data inconsistency (Mitchell 2007; Durrant-Whyte and Henderson 2008), with the goal of reducing estimation error and increasing reliability. For a comprehensive review of data fusion techniques, please refer to Castanedo (2013) and Durrant-Whyte and Henderson (2008).

At the core of probabilistic methods lies the Bayes estimator, which enables fusion of data based on Bayes rule, hence the name "Bayesian Fusion". New sensor observations can be fused at each time, and the probability density of the state estimates is updated. Data inconsistency and uncertainty can thus be handled. In the robotics literature, Bayesian fusion has been extensively used in probabilistic 2.5D mapping. For instance, Pfaff et al. (2007) fused multiple sensors' measurements by using the sum of weighted variances of the uncorrelated data. Vidal-Calleja et al. (2013) proposed to insert spatial correlation in thickness mapping within the standard Bayesian fusion, thereby improving the estimation accuracy. However, the dense covariance matrix needs to be inverted in Bayesian fusion, which takes cubic time. Kassem et al. (2016) demonstrated that Bayesian fusion could reduce uncertainty and eliminate noises caused by the cross-talk phenomena from sensor readings.

GPs provide a natural way to integrate multiple sources of incomplete and uncertain data. An initial study was carried out by Girolami (2016), who integrated heterogeneous information within a GP classification setting for protein fold recognition. Each feature was considered independent and was represented by a separate GP, and hence a composite covariance function was defined regarding a linear sum of GP priors. Vasudevan (2012) applied data fusion to estimate the elevation of terrain, given all datasets and the respective GPs (hyperparameters) that were used to model them. They presented three state-of-art fusion approaches grounded on GP regression, and the major differences between them were how to model noises and how to set the hyperparameters. The first approach added new data to an existing GP model with a fixed noise variance

(homoscedasticity[1]). The second approach was based on the concepts of the heteroscedastic[2] GP (HGP) (Kersting et al. 2007). It treated individual terrain data sets as homoscedastic in nature but different data sets considered together form a heteroscedastic system. Data from the same entity was modelled using a single set of GP hyperparameters with just the noise parameter varying between datasets. Then the fusion problem is treated as a standard GP regression problem with data having different noise parameters. This first and second approach can be problematic when faced with data from heterogeneous sensors. To overcome this problem, the dependent GP (DGP) method (Vasudevan et al. 2011) was proposed. DGP learns separate hyperparameters and noise parameters for each heterogeneous dataset and then models correlations between the individual GPs. DGP can generate multiple, correlated outputs, while being very complex and time-consuming.

Both Bayesian fusion and GP based fusion methods can take spatial correlations into consideration, which improves the accuracy but induces high computational and storage cost. As a result, these two approaches are usually used in offline scenarios. In particular, GP fusion approach can interpolate at the unobserved locations while Bayesian fusion cannot.

The information-form Bayesian fusion together with Gaussian variables were applied by Thompson et al. (2011) for efficient estimation of large scale terrain. They focused on a hierarchical distributed system in which each node estimates a subset of its parent's region, with the top-level node estimating a terrain map of the whole area. This method used a pre-specified regular finite-element mesh and the elevations of the mesh vertices were the estimated state variables. The smoothness term was applied to maintain the smoothing properties and allow for interpolation in unobserved regions. In the information form, Bayesian fusion are additive and the information matrix remains sparse, enabling constant-memory fusion of observations, efficient distribution among multiple sensing platforms and efficient solving for the estimates and uncertainty.

[1]Homoscedasticity is a property of a set of random variables where each variable has the same finite variance.

[2]Heteroscedasticity is the property of a series of random variables of not every variable having the same finite variance.

## 2.4 Efficient Approximation Methods

As was mentioned in Section 2.2.2 and 2.3, both GPs mapping and Bayesian fusion with spatial correlations have high memory consumption and computational cost, which inspired a number of approximation methods. The most related approximate methods are discussed as follows.

### 2.4.1 Gaussian Processes Approximations

In machine learning field, various methods have been proposed to approximate GP regression with Gaussian exact inference (Quiñonero-Candela and Rasmussen 2005; Rasmussen and Williams 2006; Sun et al. 2012; Chalupka et al. 2013).

The most commonly studied may be the subset methods, in which the training data, maybe the test data as well, are divided into subsets. Then GP training and prediction are done locally. The local estimates are integrated together to get the global result. An example of this is the study by Kim and Kim (2013, 2014), in which the global map was divided into overlapping submaps, and one local GP was trained for one submap. Moore and Russell (2015) developed the Gaussian Process Random Field (GPRF), which is in fact a fully connected undirected graphical model, where each node represents a submap. However, computation is still costly since the spatial correlations between each two submaps are coupled to compute the global map. However, the subset methods only preserve short length-scale structure within each submap. Moreover, choosing a submap raises questions concerning sensitivity to the choice of submaps and limitations in estimating fine-scale structure in regions that are not well covered by the submap.

Pseudo-input approaches reduce the training cost by selecting a particular subset which can represent the whole training dataset (Snelson and Ghahramani 2006). Quiñonero-Candela and Rasmussen (2005) has presented a unifying view of pseudo-input methods. Common to all the pseudo-input approximation methods is that only a subset of the underlying function values of a smaller size than the real ones are treated exactly, and the remaining latent values are approximated with cheaper computational demand. The critical and challenging issues include how to decide the number of pseudo-inputs and how to select pseudo-inputs. The choice of the number of pseudo-inputs is governed by computational cost and sensitivity to the choice.

The sparse covariance methods are developed upon the fact that the spatial correlation, computed by isotropic kernel functions, decreases gradually to almost zero as the distance between two points increases. These methods set the actual cross-covariances that are beyond a certain range to zero while ensuring the covariance matrix to be positive definite. As a consequence, a sparse covariance matrix is obtained, and the efficient sparse solvers can be employed for inference. The sparse covariance methods can be classified into two kinds, concerning how the covariance matrix is constructed. One is to design a particular kernel function which naturally generates a sparse covariance matrix, as per Melkumyan and Ramos (2009). The other is to approximate a given covariance model by multiplying the covariance function with some compactly supported taper function. (Furrer et al. 2006; Bolin and Wallin 2016). One of the major drawbacks of the sparse covariance methods is that the approximated structures may limit modelling flexibility. Another drawback is that the problems that depend on the spatial structure of locations may happen since the original model has changed.

Other representative GPs approximation methods include but are not restricted to: low-rank approaches (Williams and Seeger 2001; Banerjee et al. 2008), *Laplace inference* (Williams and Barber 1998) or *expectation propagation* inference (Minka 2001) methods, stochastic variational inference methods (Hensman et al. 2013; Gal et al. 2014). Sometimes more than one approximations are used together. For example, Ramos and Ott (2015) built *Hilbert maps*, by using fast kernel approximations that project data into a Hilbert space where a logistic regression classifier was learnt; then, the stochastic gradient approach was used to optimise the hyperparameters efficiently.

### 2.4.2 Bayesian Committee Machine

Bayesian Committee Machine (BCM) (Tresp 2000; Schwaighofer and Tresp 2003) is a typical mixture-of-expert method. BCM can be used together with GPs: local GPs take place over exclusively non-overlapping test regions, and predictions by various GPs are combined and weighted by the inverse covariance of each local prediction. An intuitively appealing effect of this weighting scheme is that modules which are uncertain about their predictions are automatically weighted less than modules that are certain about their predictions. Tresp (2000) randomly partitioned training data into subsets, while Schwaighofer and Tresp (2003) found that pre-processing

the training data with a simple clustering algorithm can lead to a drastic reduction of error rates. Note that the clustering does not make use of the target values, only the inputs.

Kim et al. (2011) firstly used BCM with GPs for occupancy mapping. Each occupancy submap was firstly predicted via local GPs. Then the predictions of the shared parts between submaps were combined via BCM. In online 3D occupancy mapping, BCM can be used to recursively update submaps with sequential observations, as has been shown by Kim and Kim (2014). More recently, Wang and Englot (2016) used BCM twice for online 3D occupancy mapping. BCM was firstly used to fuse local GP predictions together, and then to fuse new sensor observations into the existing occupancy map. Also, BCM can also be used to fuse GP occupancy submaps repeatedly with the global map during robot exploration (Jadidi et al. 2014). However, Jadidi et al. (2014) and Wang and Englot (2016) ignore the cross-correlations of test subsets to accelerate the algorithms for online mapping, which may increase estimation error.

BCM can avoid the pitfalls of subsets approximation methods, by averaging the predictions from all local GPs. BCM is formally equivalent to a pseudo inputs model in which the test points are the inducing points; i.e. it assumes that the training subsets are conditionally independent given the test data. However, the BCM approximation methods mentioned above may not reduce the computational cost. The main reason lies in the nature of BCM: it requires inverting the covariance matrices of all local GPs when fusing local estimates to predict each submap. For instance, in Kim and Kim (2014) and Wang and Englot (2016), the spatial size of each training subset, i.e. extended block, is defined to be 26 times bigger than that of the test subset, and inverting such a big covariance matrix of the training subset is costly.

The BCM approach is *transductive*[2] (Vapnik 1995) rather than inductive, in the sense that the method computes a test-set dependent model making use of the test set input locations. The BCM approximation is calculated when the inputs to the test data are known. In contrast, inductive methods, such as the pseudo-input, sparse covariance and low-rank approaches introduced in Section 2.4.1, build a model solely on basis of information from the training data.

[2]Originally, the differences between transductive and inductive learning where pointed out in statistical learning theory (Vapnik 1995). Inductive methods minimize the expected loss over all possible test sets, whereas transductive methods minimize the expected loss for one particular test set.

### 2.4.3 Submapping Techniques

Submapping strategies have become interesting approaches in robotic mapping since they work in small regions of the environment to reduce the computational cost. In the majority of cases, independent submaps of limited size are considered (Williams et al. 2002; Huang et al. 2008b; Paz et al. 2008b). Independent submaps can be consistently joined using a map joining algorithm (Tardós et al. 2002) with joining cost of $\mathcal{O}(n^2)$, or using divide and conquer method (Paz et al. 2008b) with amortised linear cost. A severe limitation with independent submaps is that they ignore the correlations between each other submap, thus producing an approximate solution and the estimates will tend to be inconsistent and inaccurate. Besides, valuable information present in one submap cannot be applied to improve the estimation of other submaps.

To address these issues, Piniés and Tardós (2008) assumed Conditional Independence, (CI), between submaps. Submaps that share common parts are firstly built, and the correlations between submaps can be obtained while recovering the global map. Kim and Kim (2013, 2014) also built submaps that share common parts, named "extended block". However, these submaps are independent submaps and information cannot be transmitted between them.

### 2.4.4 Tree data structures

The tree data structure can be used to achieve the computational gain in large-scale robotic mapping, since the tree stores and accesses data efficiently. Bertram et al. (2003) firstly used a quadtree-like clustering of the measurements into subsets, and then locally fitted them by a continuous B-spline surface. However, data correlations are not considered. Vasudevan et al. (2009) used KD-tree, a space-partitioning data structure, to search for training data near each test point. The training dataset was then approximated by the subset that lay within a specified range of the test point. Regression was performed for each test point in the centre of a corresponding sliding window. Wang and Englot (2016) proposed a new test-data octree to prune back and condense similarly-valued test data. This is on account of the fact that in occupancy mapping, many nodes in the octree share the same state class and thus can be pruned to reduce memory cost.

### 2.4.5 Markovian Approximation

Another intuitive idea is to use the Markov property to reduce the range of correlations, namely only correlations within a limited range of a region are computed. GMRFs can be regarded as global approximations to GPs, since some cross-correlations are ignored. A GMRF comes with a sparse information matrix, which allows the use of sparse matrices techniques for efficient computation. Unfortunately, GMRF suffers from a number of pitfalls. One of the major difficulties is how to specify the information matrix such that the corresponding covariance function is similar to some commonly used covariance function for a given dataset. This problem has been investigated by Rue and Tjelmeland (2002), and they proved that: for data observed on regular grids in 2D, for a large family of covariance functions, GPs can be well approximated by GMRFs with small neighbourhood structures. However, the standard MRFs are defined on grids and it is still unclear how to construct the information matrix for data that are not organised on regular grids. On the other hand, a number of approaches applied the grid-based GMRFs for non-grid data. For instance, the nearest neighbour mapping from data locations to grid locations, as per Hrafnkelsson and Cressie (2003), assumed that the GMRF values for non-grid points were equal to the closest grid cells (Wikle et al. 1998). Hartman (2006) used linear interpolation of the GMRFs grid values to assign values to non-grid locations. Although these approaches are straightforward, accuracy cannot be ensured. In recent years, Lindgren et al. (2011) proposed a continuous Markov representation of the latent Gaussian field, which has an explicit link with GPs. The most important implication of the work is that it provides a spatially consistent method for approximating continuous GPs using GMRFs.

## 2.5 Links to the Proposed Work

This section discusses the relations between some of the previous work discussed above and this thesis.

Considering that spatial correlations are essential concerning both sensor data modelling and probabilistic fusion (see Section 2.2 and 2.3), the three proposed methods in this thesis include spatial correlations during the whole mapping process. In particular, they aim for efficiently approximating the optimal global GP mapping method, named GPBF, by Vidal-Calleja et al. (2013).

subGPBF models the spatially correlated and noisy sensor data using GP regression, which is the same as GP mapping approaches (see Section 2.2.2). To address the scalability problem of GPBF, subGPBF combines GP with a novel CI submapping method. The concept of "CI submaps" has been proposed by Piniés and Tardós (2008). In particular, the backward update algorithms in subGPBF are directly inspired by their "back-propagation" algorithm, which is used for recovering the global map after loop closures. In the back-propagation algorithm, information flows in one direction starting with the most recent local submap and is propagated back through the chain of previous submaps. However, the primary limitation for the problem at hand is that: there is no explicit transition model as in a SLAM system. As a result, submap initialisation is a challenging problem when the correlation has to be learned in a map. SubGPBF addresses this problem by designing the forward update algorithm, which can initialise the following submap and ensure it contains all the currently available information.

GMRF-BF and subGMRF-BF investigate the use of GMRF models (see Section 2.2.3 and 2.4.5) and spatial correlations are modelled using sparse information matrices. A continuous GMRF representation is built using the GMRF-SPDE model proposed by Lindgren et al. (2011). GMRF-BF is a global mapping approach. It does almost all computations in information form, thereby reducing memory consumption and computational burden. subGMRF-BF speeds up GMRF-BF by imposing the CI property between submaps, based on which the information-form forward and backward update algorithms have been developed. Significant computational gain is achieved by (1) using the sparse matrix techniques and (2) by exploiting the Gaussian distribution property with the CI property. Both GMRF-BF and subGMRF-BF are significantly faster and cheaper than the original GP based 2.5D mapping approaches, such as Vidal-Calleja et al. (2013).

SubGPBF and subGMRF-BF are similar to the submapping approaches in applying a divide-and-conquer strategy but differ in that the spatial correlations between nearby submaps are saved and updated. These two methods are more efficient than GPRF (Moore and Russell 2015), which couples the correlations of every two submaps for recovering the optimal global map. Meanwhile, their main difference with Kim and Kim (2013, 2014) is that the latter build overlapping but marginally independent submaps, and information transmission between submaps is infeasible. Besides, how to use GP regression to build each submap is different. Kim and Kim (2013, 2014) recursively updates each submap with every one of its adjacent submaps, and each submap can have at most 26 neighbours in 3D space. In contrast, subGPBF and subGMRF-BF only update

each submap at most three times, during Bayesian fusion, forward update and backward update. As a result, the proposed methods in this thesis take less computational time than that of Kim and Kim (2013, 2014).

The three proposed methods are all inductive, i.e. model training is not dependent on the test data, although they seem to be similar with BCM in that the locations of test data are known. Their models, once learned, can be applied to arbitrary test points. It is a common practice in many other works regarding GP for Robotics mapping, such as (Plagemann et al. 2008a; Vasudevan et al. 2009; Wang and Englot 2016), to train a model and then use it for prediction at unobserved locations.

Concerning data fusion, GPBF and subGMRF-BF perform Bayesian fusion within submaps, and GMRF-BF and subGMRF-BF apply the information-form Bayesian fusion. They resemble terrain mapping in Pfaff et al. (2007) in that Bayesian fusion is also used, yet Pfaff et al. (2007) considers only the variance of every single value. The proposed methods consider all the auto-covariances and cross-covariances during data fusion, which is similar to the GPs based data fusion approach by Vasudevan (2012). However, Vasudevan (2012) formulated data fusion as a GP regression problem, which requires much higher computational and memory cost. In addition, while the second and third method in Vasudevan (2012), which borrow the idea from HGP (Kersting et al. 2007), can handle heteroscedastic data from homogeneous or heterogeneous sensors, this thesis considers only fusing homoscedastic datasets (the noise variance within each dataset is fixed and assumed to be known), from either homogeneous or heterogeneous sensors.

Both subGMRF-BF and Thompson et al. (2011), which was introduced in Section 2.3, are information (inverse-covariance) methods for efficient large-scale terrain mapping. Both approaches take advantages of the sparsity of information matrices, the constant-memory Bayesian fusion of Gaussian variables and the efficient mean recovery. Both approaches use finite element analysis to represent the terrain surface during prior mapping, yet the mesh modelling in subGMRF-BF is more flexible and powerful than that in Thompson et al. (2011). On the other hand, subGMRF-BF models and fuses observations at terrain surface points, while Thompson et al. (2011) estimated and fused data at the mesh vertices. In Thompson et al. (2011), the size of the system to be solved was roughly constant for a given area of a given resolution, yet at the cost of increased estimation error. To reduce the computational cost, subGMRF-BF builds CI submaps while Thompson et al.

(2011) applied a decentralised data fusion algorithm (Durrant-Whyte and Stevens 2001). Besides, subGMRF-BF can obtain consistent estimation in the boundary areas between submaps while Thompson et al. (2011) cannot (due to the independence of local nodes).

## 2.6 Summary

This chapter first briefly introduced the SLAM problem, which degenerates to the Mapping problem when robot's poses are known. This chapter then reviewed some of the influential contributions that accounted for spatial correlations, data uncertainty and large-scale dataset in the context of probabilistic robotic mapping from the following aspects: (1) GPs and GMRFs for mapping; (2) the Bayesian approaches for data fusion; (3) efficient approximations to GP and Bayesian fusion in detail. Moreover, the motivations for the work in this thesis, and its relationship with existing research were explained in detail.

# Chapter 3

# Background Theories and Techniques

This chapter establishes the mathematical notation as well as the necessary background theories and techniques used throughout this thesis. In addition, GPBF, the benchmark method for this thesis, is introduced.

## 3.1 Symbols and Notation

Matrices are capitalised, e.g. $X$. Column-wise vectors are denoted in lower case bold type, such as $\mathbf{x}$ and $\boldsymbol{\xi}$. $1:n$ means integers from 1 to $n$. $x_i$ indicates a reference to the $i$-th element of the vector $\mathbf{x}$, and $\mathbf{x}_A$ refers to a set of elements whose indices are defined by $A$. With a slight variation of notion, only in Section 3.2, $X$ represents a random variable, and $x$ represents its realisation. In particular, for GP regression, $X$ denotes the training input matrix, and $X^*$ represents the query input matrix. The $n$-by-$n$ identity matrix is denoted by $I_n$. The Euclidean norm is shown by $\|\cdot\|$, and $|x|$ represents the absolute value. The calligraphic letter $\mathcal{N}$ represents a Gaussian distribution in the covariance form, and $\mathcal{N}_c$ represents its information form. The symbol $\mathbb{E}[\cdot]$, $\mathbb{V}[\cdot]$, and $\mathbb{Cov}[\cdot]$ denote the expected value, variance, and covariance, respectively. A superscript symbol of $^\top$ denotes the transpose operation. The superscript minus sign $^-$ indicates the prior estimate, and the superscript plus sign $^+$ indicates the updated, or posterior estimate, e.g. $\boldsymbol{\mu}^+$.

Some terms that are not distinguished in general and used interchangeably in this thesis include (1) test points and query points; (2) Markov property and CI property; (3) observations and measurements; and (4) some other interchangeable concepts stated in Section 3.2.

## 3.2 Probability Theory Preliminary

This section briefly reviews some basics of probability theory that are used in this thesis. The concepts and description follow Murphy (2012).

### 3.2.1 Some Basic Concepts and Rules

Suppose $X$ is a continuous random variable, i.e. $X$ can take an uncountable number of values from a finite or countably infinite set $\mathcal{X}$. The probability of the event that $X = x$ is represented by $p(X = x)$, or just $p(x)$ for short. Here $p()$ is called the *Probability Density Function (pdf)*, or *density*, which satisfies the properties that $0 \leq p(x) \leq 1$ and $\int_{-\infty}^{\infty} p(x)dx = 1$. Let $Y$ be another continuous random variable, which can take an uncountable number of values from a finite or countably infinite set $\mathcal{Y}$.

The *joint distribution*, or *joint probability*, of the event that $X = x$ and $Y = y$ is denoted as $p(x, y)$. The *conditional probability*, or *conditional distribution*, of the event that $X = x$ given $Y = y$ is defined as

$$p(x|y) = \frac{p(x,y)}{p(y)}, \quad \text{if } p(y) > 0, \tag{3.1}$$

where | is the conditioning bar. Given the joint distribution and the conditional distribution, the *marginal distribution* is defined as

$$p(x) = \int_{y \in \mathcal{Y}} p(x,y)dy = \int_{y \in \mathcal{Y}} p(x|y)p(y)dy. \tag{3.2}$$

Based on the definitions defined above, the *product rule* is

$$p(x,y) = p(x|y)p(y). \tag{3.3}$$

The product rule can be applied multiple times to yield the *chain rule* of probability:

$$p(x_{1:n}) = p(x_1)p(x_2|x_1)p(x_3|x_2,x_1)p(x_4|x_3,x_2,x_1)\dots p(x_n|x_{1:n-1}), \tag{3.4}$$

where the notation $x_{1:n-1}$ denotes the set $\{x_1, x_2, \dots, x_{n-1}\}$.

Based on the definitions stated above, *Bayes rule*, also known as *Bayes Theorem*, is

$$p(x|y) = \frac{p(x,y)}{p(y)} = \frac{p(x)p(y|x)}{p(y)}. \tag{3.5}$$

When there are three random variables $X$, $Y$, and $Z$, Bayes rule also applies. Bayes rule can relate the *prior*, $p(x|z)$, and the *likelihood*, $p(y|x,z)$, thus giving the posterior:

$$p(x|y,z) = \frac{p(y|x,z)p(x|z)}{p(y|z)}. \tag{3.6}$$

The *mean*, or *expected value*, denoted as $\mu_x$, of $X$ with respect to $p(x)$ can be expressed through the following integral:

$$\mu_x = \mathbb{E}[x] = \int_{\mathcal{X}} xp(x)dx. \tag{3.7}$$

The *conditional expectation*, denoted as $\mu_{x|y}$, with respect to a conditional distribution $p(x|y)$ is

$$\mu_{x|y} = \mathbb{E}[x|y] = \int_{\mathcal{X}} xp(x|y)dx. \tag{3.8}$$

The *variance* of $X$ is

$$\mathbb{V}[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2. \tag{3.9}$$

The *covariance* of $X$ and $Y$ is

$$\mathbb{C}\text{ov}[x,y] = \mathbb{E}[x,y] - \mathbb{E}[x]\mathbb{E}[y]. \tag{3.10}$$

In the case of two random vectors, $\mathbf{x}$ and $\mathbf{y}$, the covariance is a matrix, denoted as $\Sigma$, and

$$\Sigma = \mathbb{C}\text{ov}[\mathbf{x},\mathbf{y}] = \mathbb{E}[\mathbf{x},\mathbf{y}][\mathbf{x}\mathbf{y}^\top] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{y}^\top]. \tag{3.11}$$

### 3.2.2 Multivariate Normal Distribution

"The **multivariate Gaussian** or **multivariate normal (MVN)** is the most widely used joint probability density function for continuous variables" (Murphy 2012, pp. 46). The pdf of the MVN on $d$-dimensional random vector $\mathbf{x}$ is defined by the following:

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right), \tag{3.12}$$

where the mean state vector $\boldsymbol{\mu} = \mathbb{E}[\mathbf{x}] \in \mathbb{R}^d$ and the positive definite covariance matrix $\Sigma = \mathbb{C}\mathrm{ov}[\mathbf{x}] \in \mathbb{R}^{d\times d}$. The representation as (3.12) is in the *covariance form*, or the *moment form*.

The MVN can also be represented in the *information form*, or the *canonical form*, which is

$$\mathbf{x} \sim \mathcal{N}_c(\boldsymbol{\eta}, Q) = (2\pi)^{-d/2}|Q|^{1/2} \exp\left(-\frac{1}{2}(\mathbf{x}^\top Q\mathbf{x} + \boldsymbol{\eta}^\top Q^{-1}\boldsymbol{\eta} - 2\mathbf{x}^\top \boldsymbol{\eta})\right), \tag{3.13}$$

where $\boldsymbol{\eta} = \Sigma^{-1}\boldsymbol{\mu}$ denotes the *information vector*, and $Q = \Sigma^{-1}$ represents the *information matrix*, or *precision matrix*. The canonical representation for the MVN is the dual of the covariance form in the sense of the fundamental processes of marginalisation and conditioning, as exemplified in Theorem 3.1 to 3.4. In particular, conditioning is easier in the information form, and marginalisation is simpler in the covariance form.

Given $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2]$ and $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma) = \mathcal{N}_c(\boldsymbol{\eta}, Q)$. Suppose the mean, covariance, information vector and information matrix of this pdf can be partitioned as follows:

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \qquad \Sigma = \begin{bmatrix} \Sigma_{11}, \Sigma_{12} \\ \Sigma_{21}, \Sigma_{22} \end{bmatrix}, \qquad \boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \end{bmatrix}, \quad Q = \begin{bmatrix} Q_{11}, Q_{12} \\ Q_{21}, Q_{22} \end{bmatrix}. \tag{3.14}$$

**Theorem 3.1.** Marginalisation in the covariance form (Murphy 2012, pp. 111).

$$p(\mathbf{x}_1) = \mathcal{N}(\boldsymbol{\mu}_1, \Sigma_{11}), \tag{3.15}$$

$$p(\mathbf{x}_2) \sim \mathcal{N}(\boldsymbol{\mu}_2, \Sigma_{22}). \tag{3.16}$$

**Theorem 3.2.** Conditioning in the covariance form (Murphy 2012, pp. 111).

$$p(\mathbf{x}_1|\mathbf{x}_2) = \mathcal{N}(\boldsymbol{\mu}_{1|2}, \Sigma_{1|2}), \quad \text{where} \tag{3.17}$$

$$\boldsymbol{\mu}_{1|2} = \boldsymbol{\mu}_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2) \tag{3.18}$$

$$= \boldsymbol{\mu}_1 - Q_{11}^{-1}Q_{12}(\mathbf{x}_2 - \boldsymbol{\mu}_2), \tag{3.19}$$

$$\Sigma_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \tag{3.20}$$

$$= Q_{11}^{-1}. \tag{3.21}$$

**Theorem 3.3.** Marginalisation in the information form (Murphy 2012, pp. 115).

$$p(\mathbf{x}_1) = \mathcal{N}_c(\boldsymbol{\eta}_1 - Q_{12}Q_{22}^{-1}\boldsymbol{\eta}_2, Q_{11} - Q_{12}Q_{22}^{-1}Q_{21}), \tag{3.22}$$

$$p(\mathbf{x}_2) = \mathcal{N}_c(\boldsymbol{\eta}_2 - Q_{21}Q_{11}^{-1}\boldsymbol{\eta}_1, Q_{22} - Q_{21}Q_{11}^{-1}Q_{12}). \tag{3.23}$$

**Theorem 3.4.** Conditioning in the information form (Murphy 2012, pp. 115).

$$p(\mathbf{x}_1|\mathbf{x}_2) = \mathcal{N}_c(\boldsymbol{\eta}_1 - Q_{12}\mathbf{x}_2, Q_{11}). \tag{3.24}$$

### 3.2.3 Independence and Conditional Independence

$X$ and $Y$ are *independent*, or *marginally independent*, denoted as $X \perp Y$, if we can represent the joint distribution as the product of the two marginals

$$X \perp Y \iff p(X, Y) = p(X)p(Y). \tag{3.25}$$

$X$ and $Y$ are *conditionally independent* (CI) given $Z$ if and only if (iff) the conditional joint distribution can be written as the product of the two conditional marginals

$$X \perp Y|Z \iff p(X, Y|Z) = p(X|Z)p(Y|Z). \tag{3.26}$$

In addition, (3.26) can be written as a graph $X - Z - Y$, which captures that all the dependencies between $X$ and $Y$ are mediated via $Z$, as will be explained in Section 3.2.4.

### 3.2.4 Probabilistic Graphical Models

This section briefly introduces some basics about *Bayes network*, also known as *Directed Graphical Model (DGM).*

**Definition 3.5** (Graph)**.** *A graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *contains a set of* nodes*, or* vertices*,* $\mathcal{V} = 1, \ldots, V$*, and a set of* edges*,* $\mathcal{E} = (i, j) : i, j \in \mathcal{V}$*. The* neighbours *of a node is defined as the set of all immediately connected nodes. A Directed Acyclic Graph (DAG) is a directed graph with no directed cycles.*

**Definition 3.6** (Graphical Model (GM))**.** *A graphical model is a method to represent a joint distribution by making CI assumptions. Particularly, the nodes in the graph represent random variables, and the lack of edges represent CI assumptions. A Bayes network is a GM whose graph is a DAG.*

**Definition 3.7** (CI property)**.** $\xi_{\mathcal{A}} \perp \xi_C | \xi_B \iff \mathcal{A}$ *is d-separated from* $C$ *given* $B$*.*

**Definition 3.8** ((first order) Markov assumption)**.** *In a sequence of observations* $\{\xi_1, \xi_2, \ldots, \xi_t\}$*, if*

$$\xi_t \perp \xi_{1:t-2} | \xi_{t-1}\,, \tag{3.27}$$

*the conditional probability becomes*

$$p(\xi_t | \xi_{t-1}, \xi_{t-2}, \ldots, \xi_1) = p(\xi_t | \xi_{t-1})\,. \tag{3.28}$$

**Definition 3.9** (the chain rule of Bayes Network)**.** *Based on Markov assumption and the chain rule, the joint distribution can be written as follows:*

$$p(\xi_1, \xi_2, \ldots, \xi_{t-1}, \xi_t) = p(\xi_1) \prod_{i=2}^{t} p(\xi_i | \xi_{i-1})\,. \tag{3.29}$$

This is called a (first-order) *Markov chain.* They can be characterised by an initial distribution over states, $p(\xi_1)$, plus a *state transition model* $p(\xi_t | \xi_{t-1})$, $t > 2$.

For example, Figure 3.1 shows a simple Bayes network, which represents the probabilistic dependence between a set of variables $\xi_a$, $\xi_b$ and $\xi_c$. It shows that the set of nodes $a$ is d-separated from $c$ given $b$ iff each path from every node $a_i \in a$ to every node $c_i \in c$ is d-separated by $b$.

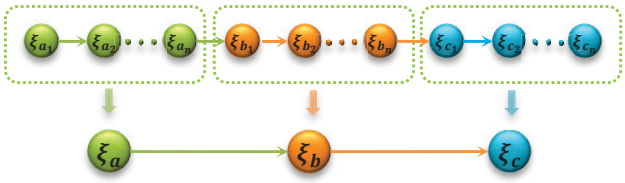


FIGURE 3.1: Schematic representation of a Bayes network showing three sets of nodes.

Consider the probability $p(\xi_a, \xi_b, \xi_c)$ encoded in Figure 3.1. When conditioning on $\xi_b$, we have

$$p(\xi_a, \xi_c|\xi_b) = \frac{p(\xi_a)p(\xi_b|\xi_a)p(\xi_c|\xi_b)}{p(\xi_b)} = \frac{p(\xi_a, \xi_b)p(\xi_c|\xi_b)}{p(\xi_b)} = p(\xi_a|\xi_b)p(\xi_c|\xi_b),$$

and thus $\xi_a \perp \xi_c|\xi_b$ (see (3.26)). Note that the set of nodes $b$ breaks the chain into two, as in a Markov chain.

Consider the nodes in Figure 3.1 represent a set of Gaussian variables, and take nodes $\xi_{a_1}, \ldots, \xi_{a_n}$ as an example. For $i \neq j$, node $\xi_{a_i}$ is correlated with node $\xi_{a_j}$, thus the covariance matrix of $p(\xi_{a_1}, \ldots, \xi_{a_n})$ is dense. Meanwhile, for $i \neq j$, $\xi_{a_i}$ and $\xi_{a_j}$ are *conditionally independent* given the rest of nodes. Therefore, $Q_{\xi_{a_i}, \xi_{a_j}} = 0$ for $i \neq j$ and $Q$ has most of the elements being zero, where $Q$ is the information matrix of $p(\xi_{a_1}, \ldots, \xi_{a_n})$. This shows that *while the covariance matrix models the independence of variables, the information matrix encodes the conditional independence*; in fact, the CI property is naturally reflected in the sparse pattern of the information matrix.

### 3.2.5 Linear Gaussian Systems

Suppose we have two random variables $\mathbf{x} \in \mathbb{R}^{d_\mathbf{x}}$ and $\mathbf{y} \in \mathbb{R}^{d_\mathbf{y}}$. Suppose $\mathbf{x}$ is a hidden variable, and $\mathbf{y}$ is the noisy observation of $\mathbf{x}$. Let us assume we have the following prior and likelihood:

$$p(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_\mathbf{x}, \Sigma_\mathbf{x}), \tag{3.30}$$

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(H\mathbf{x} + \mathbf{b}, \Sigma_\mathbf{y}), \tag{3.31}$$

where $H$ is a matrix of size $d_\mathbf{y} \times d_\mathbf{x}$. This is an example of a linear Gaussian System. The following rule can be used to infer $\mathbf{x}$ from $\mathbf{y}$.

**Theorem 3.10.** Bayes rule for linear Gaussian systems (Murphy 2012, pp. 119). *Given a linear Gaussian system, as in* (3.30) *and (3.31), the posterior distribution of* $\mathbf{x}$ *is given by*

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}|\mathbf{y}}, \Sigma_{\mathbf{x}|\mathbf{y}}), \quad \text{where} \tag{3.32}$$

$$\Sigma_{\mathbf{x}|\mathbf{y}}^{-1} = \Sigma_{\mathbf{x}}^{-1} + H^{\top}\Sigma_{\mathbf{y}}^{-1}H, \tag{3.33}$$

$$\boldsymbol{\mu}_{\mathbf{x}|\mathbf{y}} = \Sigma_{\mathbf{x}|\mathbf{y}}\left(H^{\top}\Sigma_{\mathbf{y}}^{-1}(\mathbf{y}-\mathbf{b}) + \Sigma_{\mathbf{x}}^{-1}\boldsymbol{\mu}_{\mathbf{x}}\right). \tag{3.34}$$

## 3.3 Gaussian Processes

GPs were initially formalised for machine learning tasks by Williams and Rasmussen (1996) and Neal (1996). GPs have become one of the most popular non-parametric Bayesian models which have been developed for different tasks such as density estimation, regression, classification, topic modelling. In this thesis, we will focus only on regression problems. GPs are often the preferred approach as they offer several useful properties:

- ◇ GPs provide a principled, practical, and probabilistic approach to doing inference and learning in kernel machines. The Bayesian nature allows GPs to incorporate prior knowledge and link the related sources of information. GPs also provide a measure of estimate uncertainty, taking into account the noise of the data points.
- ◇ GPs do not require a discretised representation of an environment and they are able to predict function values at arbitrary locations.
- ◇ GPs can approximate a wide range of problem domains. Instead of working over a parameter space, GPs place a prior directly on the space of functions without parametrising the function, thus being *non-parametric*. Consequently, the computational complexity of inference scales as the number of data points instead of number of parameters.

The rest of this section gives a brief introduction of GPs. The concepts and notation are mainly on the basis of Gaussian Processes for Machine Learning by Rasmussen and Williams (2006) and the book section of Gaussian Processes by Quadrianto et al. (2010).

### 3.3.1 Definition

**Definition 3.11** (GP). *A GP is a stochastic process in which all the finite-dimensional distributions are MVNs for any finite choice of variables.*

In general, GPs are used to define a probability distribution over functions $f\colon \mathcal{X} \to \mathbb{R}$ such that the set of values of $f$ evaluated at an arbitrary set of points $\{\mathbf{x}_i\}_{i=1}^N \in \mathcal{X}$ will have an $N$-variate Gaussian distribution. The set $\mathcal{X}$ is usually a subset of $\mathbb{R}^d$. When $d = 2$, a GP is also known as a *Gaussian Random Field (GRF)*.

A GP is completely specified by its mean function and covariance function. We define the mean function $m(\mathbf{x})$ and the covariance function $k(\mathbf{x}, \mathbf{x}')$ of a real process $f(\mathbf{x})$ as

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})], \tag{3.35}$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{C}\mathrm{o}\mathbb{v}[f(\mathbf{x}), f(\mathbf{x}')] = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x})(f(\mathbf{x}') - m(\mathbf{x}')], \tag{3.36}$$

and we will write the GP as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \tag{3.37}$$

The zero mean function is usually taken, as we can always centre the observed outputs to have a zero mean. The covariance matrix must be a positive definite function to ensure the existence of all finite-dimensional distribution, i.e. to ensure the positive definiteness of all covariance matrices of the finite-dimensional MVNs.

### 3.3.2 GP Regression and Model Selection

In the regression problem, we are interested in recovering a functional dependency

$$y_i = f(\mathbf{x}_i) + \epsilon = \xi_i + \epsilon, \tag{3.38}$$

given a training dataset $\Psi_1 = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $y_i \in \mathcal{Y}$ is the noisy output observed at the input $\mathbf{x}_i \in \mathcal{X}$. $\mathcal{Y}$ is a subset of $\mathbb{R}$, $\mathcal{X}$ is a subset of $\mathbb{R}^d$. $\xi_i = f(\mathbf{x}_i)$ and $f$ is the noise-free latent function.

$\epsilon$ denotes the independent and identically distributed (iid) Gaussian noise, and $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$. For 2.5D data, $\mathcal{X} \in \mathbb{R}^2$ and $y_i \in \mathcal{Y} \in \mathbb{R}$, and we will use 2.5D data in the rest of this thesis.

Then the prior distribution on the noisy observations becomes

$$\mathbb{C}\mathrm{ov}[y_i, y_j] = k(\mathbf{x}_i, \mathbf{x}_j) + \sigma_\epsilon^2 \delta_{ij}, \quad \text{or} \quad \mathbb{C}\mathrm{ov}[\mathbf{y}] = K(X, X) + \sigma_\epsilon^2 I, \tag{3.39}$$

where $\delta_{ij}$ is a Kronecker delta which is one iff $i = j$ and zero otherwise. The training input matrix $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^\top$ and $X \in \mathbb{R}^{n \times 2}$. The training output is $\mathbf{y} = [y_1; y_2; \ldots; y_n]$.

The final goal in regression is to predict the noise-free values $\boldsymbol{\xi}^* = f(X^*)$ at the (unobserved) query points $X^* = [\mathbf{x}_1^*, \mathbf{x}_2^*, \ldots, \mathbf{x}_{n^*}^*]^\top$. When a zero-mean GP prior is placed over the latent function, (3.37) is

$$f \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}')), \tag{3.40}$$

and the joint distribution over the $\mathbf{y}$ and $\boldsymbol{\xi}^*$ according to the prior is

$$\begin{bmatrix} \mathbf{y} \\ \boldsymbol{\xi}^* \end{bmatrix} \sim \mathcal{N}\left( \mathbf{0}, \begin{bmatrix} K(X, X) + \sigma_\epsilon^2 I_n & K(X, X^*) \\ K(X, X^*)^\top & K(X^*, X^*) \end{bmatrix} \right). \tag{3.41}$$

The $n \times n$ matrix $K(X, X)$ is the *covariance matrix*[1] of the training inputs, and it has the entries of $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. The $n \times n^*$ matrix $K(X, X^*)$ denotes the covariances evaluated at all pairs of training and query inputs, and has the entries of $k(\mathbf{x}_i, \mathbf{x}_j^*)$ The matrices $K(X^*, X^*)$ are defined in the similar way.

Then the predictive distribution is derived based on the conditioning property of MVNs:

$$\boldsymbol{\xi}^* | X, \mathbf{y}, X^* \sim \mathcal{N}(\boldsymbol{\mu}^*, \Sigma^*), \quad \text{where} \tag{3.42}$$

$$\mu^* = K(X^*, X)^\top [K(X, X) + \sigma_\epsilon^2 I_n]^{-1} \mathbf{y}, \tag{3.43}$$

$$\Sigma^* = K(X^*, X^*) - K(X^*, X)^\top [K(X, X) + \sigma_\epsilon^2 I_n]^{-1} K(X, X^*). \tag{3.44}$$

[1] Given a set of input points $\{\mathbf{x}_i\}_{i=1}^n$, the *Gram matrix* $K$ has the entries of $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. If $k$ is a covariance function, $K$ is called the *covariance matrix* (Rasmussen and Williams 2006, ch. 4).

For each query input $\mathbf{x}^*$, $\xi_i^* = f(\mathbf{x}^*)$ and its predictive distribution is

$$\xi_i^* | X, \mathbf{y}, X^* \sim \mathcal{N}(\mu^*, (\sigma^*)^2), \quad \text{where} \tag{3.45}$$

$$\mu^* = k(\mathbf{x}^*, X)^\top [K(X,X) + \sigma_\epsilon^2 I_n]^{-1} \mathbf{y}, \tag{3.46}$$

$$(\sigma^*)^2 = k(\mathbf{x}^*, \mathbf{x}^*) - k(\mathbf{x}^*, X)^\top [K(X,X) + \sigma_\epsilon^2 I_n]^{-1} k(X, \mathbf{x}^*). \tag{3.47}$$

Such estimator at a query input is derived as the minimum Mean Squared Error (MSE) linear predictor (see Schabenberger and Gotway (2004); Rasmussen and Williams (2006) for details). Note that the matrix inversion in this thesis, such as $K_{\mathbf{y}}^{-1} = [K(X,X) + \sigma_\epsilon^2 I_n]^{-1}$ in (3.43), is computed via solving the linear system $K_{\mathbf{y}}\mathbf{x} = I$ using the Cholesky decomposition for superior efficiency and numerical stability. The Cholesky factorisation or Cholesky decomposition is a fundamental tool in matrix computations, and it is mainly used for the numerical solution of linear equations $A\mathbf{x} = \mathbf{b}$. If $A$ is symmetric and positive definite, then we can solve $A\mathbf{x} = \mathbf{b}$ by first computing the Cholesky decomposition $\mathbf{A} = LL^\top$, then solving $L\mathbf{y} = \mathbf{b}$ for $\mathbf{y}$ by forward substitution, and finally solving $L^\top \mathbf{x} = \mathbf{y}$ for $x$ by back substitution. The standard algorithm for its computation dates from the early part of this century (Householder 1964, p.208) and it is one of the most numerically stable of all matrix algorithms (Wilkinson 1968; Kielbasinski 1987).

As explained above, GPs provide an explicit probabilistic formulation of the problem, which directly generates confidence intervals for regression. This cannot be achieved with other non-Bayesian kernel approaches. Compared with the traditional Bayesian linear regression models, which parametrise the latent function $f$ by some parameters and defines a prior distribution over the parameters, GP regression places a prior directly on the space of $f$ without parametrising it. Therefore, GPs can discover the $f$ which requires infinitely many functional forms, i.e. infinite number of parameters, whereas this cannot be solved by the traditional Bayesian linear models.

### Model Selection

In many practical applications, the functional form of the covariance function needs to be chosen, and any values of hyperparameters need to be optimally determined. This is called *model selection.* The hyperparameters $\boldsymbol{\theta}$ includes the free parameters in the mean, the covariance and the likelihood function. For extensive discussions about the model selection problem and various methods to determine the hyperparameters from training data, please refer to Rasmussen

and Williams (2006), ch. 5. One of the commonly used methods is the maximum likelihood. The logarithm of the marginal likelihood for the hyperparameters $\boldsymbol{\theta}$ is in the form of

$$\log p(\mathbf{y}|X,\theta) = -\frac{1}{2}\mathbf{y}^\top K_\mathbf{y}^{-1}\mathbf{y} - \frac{1}{2}\log|K_\mathbf{y}| - \frac{n}{2}\log 2\pi\,, \tag{3.48}$$

where $K_\mathbf{y} = K(X,X) + \sigma_\epsilon^2 I_n$ is the covariance matrix for $\mathbf{y}$. The three terms of the log-likelihood function have readily interpretable roles: the only term involving the observed targets is the data-fit $-\mathbf{y}^\top K_\mathbf{y}^{-1}\mathbf{y}/2$; $\log|K_\mathbf{y}|/2$ is the complexity penalty depending only on the covariance function and the inputs and $n\log(2\pi)/2$ is a normalisation constant. To set $\boldsymbol{\theta}$ by maximising the marginal likelihood, we compute the partial derivatives of (3.48) with respect to (w.r.t.) $\boldsymbol{\theta}$ as

$$\begin{aligned}\frac{\partial}{\partial\theta_i}\log p(\mathbf{y}|X,\boldsymbol{\theta}) &= \frac{1}{2}\mathbf{y}^\top K_\mathbf{y}^{-1}\frac{\partial K_\mathbf{y}}{\partial\theta_i}K_\mathbf{y}^{-1}\mathbf{y} - \frac{1}{2}\mathrm{tr}(K_\mathbf{y}^{-1}\frac{\partial K_\mathbf{y}}{\partial\theta_i}) \\ &= \mathrm{tr}\Big((\boldsymbol{\alpha}\boldsymbol{\alpha}^\top - K_\mathbf{y}^{-1})\frac{\partial K_\mathbf{y}}{\partial\boldsymbol{\theta}}\Big), \quad \text{where}\boldsymbol{\alpha} = K_\mathbf{y}^{-1}\mathbf{y}.\end{aligned} \tag{3.49}$$

The complexity of computing (3.48) is dominated by inverting $K_\mathbf{y}$, which typically requires time $\mathcal{O}(n^3)$. Once $K_\mathbf{y}^{-1}$ is known, computing (3.49) requires only $\mathcal{O}(n^2)$ per hyperparameter[2], so using a gradient based optimiser is advantageous (Boyd and Vandenberghe 2004; Ruder 2016).

### Computational Complexity

As stated above, if there are no simplifying assumptions on the covariance matrix, GP training takes $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ memory due to computing $K_\mathbf{y}^{-1}$. This means that with 1000 observations, the number of operations required is in the order of $10^9$, and $10^6$ allocation for memory storage. In addition, such matrix inversion is required in each iteration of the gradient descent algorithm for maximising the likelihood, rendering them infeasible for large spatial datasets, where one might have more than $10^4$ measurements. For the GP prediction, calculating the predictive mean (see (3.43)) requires $\mathcal{O}(n^3 + n^*n)$ time and $\mathcal{O}(n^2 + n^*n)$ memory; and computing the covariance (see (3.44)) takes $\mathcal{O}(n^3 + n^*n^2)$ time and $\mathcal{O}(n^2 + (n^*)^2)$ memory. Therefore, GP cannot scale well with large training data nor query data.

[2]Note that matrix-by-matrix products in (3.49) should not be computed directly: in the first term, do the vector-by-matrix multiplication first; in the trace term, compute only the diagonal terms of the product.

### 3.3.3 Covariance Function

*Covariance functions*, or *kernels*, are the cornerstone of GP, as they encode information about the correlations between measurements. The kernels are widely applicable models of an underlying process and can be tailored to specific applications by the refinement of their hyperparameters. A covariance function $k_\theta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, with the set of hyperparameters $\boldsymbol{\theta}$, computes the covariance

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{Cov}[f(\mathbf{x}), f(\mathbf{x}')] \tag{3.50}$$

of the latent function $f$ between the inputs $\mathbf{x}$ and $\mathbf{x}'$.

Covariance functions can be generally divided into two kinds: *stationary* and *non-stationary*. A stationary function is a function of $\mathbf{x} - \mathbf{x}'$, i.e. $k(\mathbf{x}, \mathbf{x}') = D(\mathbf{x}, \mathbf{x}')$ for some distance function $D$. Furthermore, if a covariance function is a function only of $|\mathbf{x} - \mathbf{x}'|$ then it is *isotropic*, otherwise it is *anisotropic*. Some typical examples include the Squared Exponential (SE), rational quadratic and Matérn family. The dot-product based function, such as the polynomial function, is one of the most widely used non-stationary covariance functions. Table 3.1 compares some common covariance functions in their isotropic forms. For their anisotropic versions, please refer to Chapter 4 in Rasmussen and Williams (2006).

Table 3.1: Examples of some covariance functions $k(\mathbf{x}, \mathbf{x}')$.

| Name | $k(\mathbf{x}, \mathbf{x}')$ | $\boldsymbol{\theta}$ | Remark |
|---|---|---|---|
| Squared exponential (SE) | $\sigma_f^2 \exp(-\frac{r^2}{2l^2})$ | $\{\sigma_f, l\}$ | Strong smoothness assumption |
| Matérn family | $\sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}r}{l}\right)^\nu K_\nu \left(\frac{\sqrt{2\nu}r}{l}\right)$ | $\{\sigma_f, \nu, l\}$ | Flexible to be smooth or not |
| Exponential | $\sigma_f^2 \exp(-\frac{r}{l})$ | $\{l\}$ | Matérn when $\nu = 1/2$ |
| Rational quadratic | $\sigma_f^2 \left(1 + \frac{r^2}{2\alpha l^2}\right)^{-\alpha}$ | $\{\sigma_f, \alpha, l\}$ | An infinite sum of SE |
| Polynomial | $\sigma_f^2 (\mathbf{x} \cdot \mathbf{x}' + \sigma_0^2)^p$ | $\{\sigma_f, \sigma_0\}$ | $p$ is a positive integer |

The distance $r = \|\mathbf{x} - \mathbf{x}'\|$. $\sigma_f^2$ is the signal variance. $l$ is the length scale. The $\cdot$ denotes the dot product. This table is adapted from Table 1 in Quadrianto et al. (2010).

The Matérn family (Stein 1999) is very flexible and general, which can generalise many of the most-used covariance functions. It has been argued by others, in particular Stein (1999), that the Matérn class is the only class of covariance function needed for practical spatial statistics. A

Matérn covariance function is given by

$$k(\mathbf{x},\mathbf{x}') = k_\nu(r) = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left( \frac{\sqrt{2\nu} r}{l} \right)^\nu K_\nu \left( \frac{\sqrt{2\nu} r}{l} \right), \tag{3.51}$$

which depends on data only through the distance $r = \|\mathbf{x} - \mathbf{x}'\|$. $\Gamma(\cdot)$ is the Gamma function. $K_\nu(\cdot)$ is the modified Bessel function of the second kind, with the order of $\nu > 0$. $\nu$ is a differentiability parameter, which controls the smoothness of the underlying process: the larger $\nu$ is, the smoother it is. $l$ is the characteristic length scale, which is a range parameter. The correlation diminishes to almost zero as the distance approaches $l$. $\sigma_f$ is the signal variance. $K_\nu(\cdot)$ has a closed form when $\nu = \alpha + 1/2$ for $\alpha$ a non-negative integer (Hodges 2013; Rasmussen and Williams 2006):

$$k_{\nu=\alpha+1/2}(r) = \exp(-\frac{\sqrt{2\nu} r}{l}) \frac{\Gamma(\alpha+1)}{\Gamma(2\alpha+1)} \sum_{i=0}^{\alpha} \frac{(\alpha+i)!}{i!(\alpha-i)!} (\frac{\sqrt{8\nu} r}{l})^{\alpha-i}. \tag{3.52}$$

For $\alpha = 0$ and $\nu = 1/2$, (3.52) is the exponential covariance function, which can model non-smooth behaviours. When $\nu = 3/2$ and $\nu = 5/2$, the Matérn kernel becomes

$$k_{\nu=3/2}(r) = \sigma_f^2 (1 + \frac{\sqrt{3} r}{l}) \exp(-\frac{\sqrt{3} r}{l}), \quad \text{and} \tag{3.53}$$

$$k_{\nu=5/2}(r) = \sigma_f^2 (1 + \frac{\sqrt{5} r}{l} + \frac{5r^2}{3l^2}) \exp(-\frac{\sqrt{5} r}{l}), \tag{3.54}$$

which are the most interesting cases for machine learning. As $\nu \to \infty$, the Matérn kernel converges to the SE kernel. The SE kernel is infinitely differentiable, thus it is very smooth. Roughly speaking, a smooth kernel is suitable to model highly correlated data.

The sparse covariance methods, as mentioned in Section 2.4.1, can be applied to taper a true covariance matrix $k(\mathbf{x},\mathbf{x}')$ to zero beyond a certain range by multiplying it with some compactly supported positive definite taper function $k_l(\mathbf{x},\mathbf{x}')$, which gives the tapered covariance

$$k_{tap}(\mathbf{x},\mathbf{x}') = k_l(\mathbf{x},\mathbf{x}') k(\mathbf{x},\mathbf{x}'), \tag{3.55}$$

*Compactly supported* means the covariance between points becomes zero when their distance exceeds the threshold. Using $k_{tap}(\mathbf{x},\mathbf{x}')$, the matrix $K_\mathbf{y}$ in the GP predictor is sparse, which may lead to computational advantages. One important issue in designing the tapered covariance functions is to guarantee their positive definiteness, yet they are usually positive definite for all input

dimensions. For instance, the *compactly-supported piecewise polynomial covariance functions* are positive definite in $\mathbb{R}^d$, where $d$ is the data dimension. An example of such kernel is

$$k_{pp,2}(r) = \sigma_f^2 \max(1-r,0)^{j+2}\Big(\big(j^2+4j+3\big)r^2 + (3j+6)\,r + 3\Big)/3\,,\ j = \left\lfloor \frac{d}{2} \right\rfloor + 3\,, \tag{3.56}$$

which is 2-times mean-square differentiable. The distance $r = (\|\mathbf{x}-\mathbf{x}'\|)/l$, $l$ is the length scale and $\sigma_f^2$ is the variance.

### 3.3.4 GP Software

Two of the best known GP software packages are *Gaussian Processes for Machine Learning (GPML)* (Rasmussen and Nickisch 2010) and *GPstuff* (Vanhatalo et al. 2013). Both GPML and GPstuff collect a versatile range of kernel models and computational methods, and they both have user-friendly interfaces. GPML does not require other toolboxes. GPstuff requires Netlab (Nabney 2002) and SuitSparse (Davis 2005). GPstuff is better known in the statistics field. GPs libraries are also available in R, e.g. *GPfit* (MacDonald et al. 2013), while *pyGPs* (Neumann et al. 2014) is developed in Python. Overviews including some alternatives are provided by the Gaussian Processes website (http://www. gaussianprocess.org/) and the R Archive Network (http://cran.r-project.org/). This thesis uses GPML for the implementation.

## 3.4 Gaussian Markov Random Fields and the SPDE approach

This section presents a brief introduction of the GMRFs. The concepts and notation follow Gaussian Markov random fields: theory and applications by Rue and Held (2005) and the original paper for the GMRF-SPDE approach by Lindgren et al. (2011).

### 3.4.1 Definition

A GRF is one of a few appropriate multivariate models with an explicit and computable normalising constant and has excellent analytic properties. In Section 3.3, a GP considers the input domain $\mathcal{X}$ to be continuous, typically a subset of $\mathbb{R}^d$. When considering a discrete $\mathcal{X}$, e.g. lattices (regular

grids), one popular choice is to use GMRFs to analyse the measured data. A GMRF combines a MVN with an additional conditional independence property, which enables computational gain.

**Definition 3.12** (GMRF)**.** *A random vector, $\boldsymbol{\xi} = [\xi_1, \xi_2, \ldots, \xi_n]^\top$, defined over a set of discretely indexed sites $i = 1, \ldots, n$, $i \in \mathcal{X}$ is called a GMRF w.r.t. the graph $\mathcal{G} = (\mathcal{V} = 1, \ldots, n, \mathcal{E})$ with mean $\mu$ and information matrix $Q$, iff*

$$\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{\mu}, Q^{-1}), \tag{3.57}$$

*and its pdf has the form of*

$$p(\boldsymbol{\xi}) = (2\pi)^{-n/2} |Q|^{1/2} exp\left(-\frac{1}{2}(\boldsymbol{\xi} - \boldsymbol{\mu})^\top Q(\boldsymbol{\xi} - \boldsymbol{\mu})\right), \tag{3.58}$$

*and $Q_{ij} \neq 0 \iff i, j \in \mathcal{E}$ for all $i \neq j$.*

Here (3.58) can be regarded as a MVN written by the parameters of mean $\boldsymbol{\mu}$ and information matrix $Q$ (see (3.13)). In a GMRF, $Q$ encodes the CI property (see Definition 3.7), or Markov property, which is depicted as

$$\text{for some } i \neq j\,,\ \xi_i \perp \xi_j | \boldsymbol{\xi}_{-i,j} \iff Q_{ij} = 0 \iff j \notin \mathcal{N}_i\,, \tag{3.59}$$

where $\boldsymbol{\xi}_{-i,j}$ denotes all the elements in $\boldsymbol{\xi}$ except $\xi_i$ and $\xi_j$, and $\mathcal{N}_i$ is the neighbourhood of $i$. This means that the following properties are equivalent:

- $\xi_i$ and $\xi_j$ are conditionally independent;
- the associated entry in the information matrix, denoted as $Q_{ij}$, equals to zero; and
- location index $i$ and $j$ are not neighbours.

The zeros entries in $Q$ are called *structural zeros*, since they represent the absent edges in the GM. The neighbourhood of $i$ typically consists of all points that, in some sense, are close to $i$. In theory, there are no restrictions on the size of the neighbourhood, and one could, for example, have $\mathcal{N}_i$ include all elements in $\boldsymbol{\xi}$ except $\xi_i$. However, the advantages of the Markov assumption naturally occurs when the neighbourhood is small. In fact, since $Q_{ij} \neq 0$ only if $i$ and $j$ are neighbours, usually only $\mathcal{O}(n)$ of the $n^2$ entries of $Q$ are non-zero, therefore, $Q$ has a very low

*density*[3] and is a sparse matrix. This facilitates the usage of the efficient techniques for sparse matrix operations when working with GMRFs, as will be discussed in Section 5.2.3.

### 3.4.2 GMRF Regression and the SPDE approach

Traditionally, Markov models have been mostly confined to discretely indexed spatial domain $\mathcal{X}$. In the case when the measurements were observed at irregularly-spaced locations, one often interpolates the locations to a regular grid first before applying the GMRF model, as in Section 2.4.5. Such interpolation inevitably loses information. Besides, the resolution of the grid has a significant impact on the inference. Particularly, the traditional GMRF models cannot learn the spatial correlations of non-regularly located data.

On the other hand, Whittle (1963) and Lindgren et al. (2011) advocate that one can view a large class of random field models, including a GRF with a Matérn kernel, as a solution to the continuous domain SPDE. Lindgren et al. (2011) has derived a method for explicit and efficient Markov representations of the Gaussian Matérn fields. The method is based on the fact that a random process on $\mathbb{R}^d$ with a Matérn kernel is a solution to the SPDE. Therefore the Markov representation is obtained by considering an approximate stochastic weak solution to the SPDE, without defining a Matérn field through a covariance function.

When dealing with Bayesian inference for GMRF regression, the Integrated Nested Laplace Approximation (INLA) algorithm (Rue et al. 2009) is an alternative to the Markov chain Monte Carlo (MCMC) method, while having additional computational advantages (Rue et al. 2017). The INLA algorithm was implemented in an R package named INLA (Martino and Rue 2010). Later, INLA was combined with the SPDE approach in order to account for point-reference data, which has been implemented in the R-INLA package (Lindgren and Rue 2015). Therefore, the GMRF-SPDE approach can be implemented efficiently to solve regression problems.

Considering the same regression problem as in Section 3.3.2, the MVN distribution of the latent function can be written as

$$\xi \sim \mathcal{N}(0, Q^{-1}(X,X)), \quad \text{where } Q^{-1}(X,X) = K(X,X). \tag{3.60}$$

[3]The total number of elements is called the *density* of a matrix.

Since a zero mean function is used, $\boldsymbol{\xi}$ is specified by $Q$. R-INLA approximates $\xi$ with piecewise linear basis functions that are defined on a triangular domain, and then gets the the hyperparameters, including the noise variance $\sigma_\epsilon^2$, of the GMRF by solving the SPDE using finite element methods (see Lindgren et al. (2011) and Krainski et al. (2017) for details). With the learned hyperparameters, $Q$ can be computed using the methods described in Appendix E.

The predictive distribution at input $X^*$ has the following information matrix and mean vector:

$$Q^* = Q(X^*, X^*) + Q_\epsilon\,, \tag{3.61}$$

$$\boldsymbol{\mu}^* = (Q^*)^{-1} Q(X^*, X)\mathbf{y}\,. \tag{3.62}$$

Here $Q_\epsilon$ is a diagonal matrix that represents the precision of data, i.e. the inverse of the variance of data noise. The information vector $\boldsymbol{\eta}^*$ that corresponds to $\boldsymbol{\mu}^*$ becomes

$$\boldsymbol{\eta}^* = Q^*\boldsymbol{\mu} = Q(X^*, X)\mathbf{y}\,. \tag{3.63}$$

The comparison between (3.62) and (3.63) shows that maintaining $\boldsymbol{\eta}^*$ is cheaper than estimating its dual. When $\boldsymbol{\mu}^*$ is needed, the expensive calculation of $(Q^*)^{-1}Q(X^*, X)\mathbf{y}$ in (3.62) can be done by applying the Cholesky factorisation, which was briefly introduced in Section 3.3.2. We first compute $Q^* = LL^\top$ and then solve $Q^*\mathbf{x} = Q(X^*, X)\mathbf{y}$ using forward and backward substitutions.

The practical significance of this GMRF-SPDE approach is that classical GRFs can be merged with methods based on the Markov property, providing continuous domain models that are computationally efficient, and where the parameters can be specified locally without having to worry about positive definiteness of covariance functions. Additionally, Lindgren and Rue (2015) proved that the GMRF-SPDE approach can generalise to a large class of covariance functions, including oscillating and non-stationary, without needing explicitly to derive a covariance function.

### 3.4.3 GMRF Software

The R-library *R-INLA* (refer to http://www.r-inla.org/ and Lindgren and Rue (2015)) is the only implementation interface for the GMRF-SPDE approach. It covers stationary spatial models, non-stationary spatial models, and also spatiotemporal models. R-INLA is user-friendly, and has been

successfully applied in epidemiology, ecology, environmental risk assessment, as well as general geostatistics. Rue et al. (2017) points out that "R-INLA has turned out to be very popular in applied science and applied statistics, and has become a versatile tool for quick and reliable Bayesian inference". There are some other packages for the GMRF with Bayesian inference, without the SPDE approach. For example, *LatticeKrig* package in CRAN (Nychka et al. 2013) and *GMRFLib* in C (Rue and Follestad 2001). This thesis uses R-INLA for GMRF regression.

## 3.5 Bayesian Data Fusion for Linear Gaussian Systems

Bayesian Fusion provides a direct method of combining the observed information with prior beliefs about the state of random variables.

### 3.5.1 Covariance-form Bayesian Fusion

This section considers Bayesian fusion of two sets of noisy measurements. Assume two sets of noisy observations $\mathbf{y} = \{y_i\}_{i=1}^{n_y}$ and $\mathbf{z} = \{z_j\}_{j=1}^{n_z}$ have been taken from two different sensing sources. Let $\mathbf{x} = \{x_i\}_{i=1}^{n_y}$ denote the noise-free latent variables to be estimated. Define the observation models to be

$$y_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma_i), \tag{3.64}$$

$$\mathbf{z} = H\mathbf{x} + \boldsymbol{\delta}, \quad \boldsymbol{\delta} \sim \mathcal{N}(0, R). \tag{3.65}$$

In (3.64), $f(\cdot)$ denotes a latent process (can be linear or non-linear) which projects from the estimated state variable $x_i$ to the noisy observation $y_i$ when a Gaussian white noise $\epsilon_i$ is added. A linear observation model with Gaussian relationships is defined in (3.65), where $H$ denotes the observation matrix of size $n_z \times n_y$; the observed variable $\mathbf{z}$ is in the same space as the estimated state $\mathbf{x}$ with $H = I_{n_z \times n_y}$. The measurement noise $\boldsymbol{\delta}$ can be assumed to be a zero-mean Gaussian with covariance $R$. If no correlation is included, $R$ is a diagonal matrix which contains noise variances. Measurement noise are often known in advance in reality. In (3.64), standard regression methods can be applied to discover the latent variables $\mathbf{x}$ from the observations $\mathbf{y}$, thus we can

define

$$p(\mathbf{x}|\mathbf{y}) \triangleq \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}}, \Sigma_{\mathbf{x}}). \tag{3.66}$$

According to the linear model in (3.65), the likelihood function $p(\mathbf{z}|\mathbf{x})$ is given by

$$p(\mathbf{z}|\mathbf{x}) = \mathcal{N}(H\mathbf{x}, R). \tag{3.67}$$

In Bayesian fusion, the aim is to estimate $p(\mathbf{x}|\mathbf{y}, \mathbf{z}) \triangleq \mathcal{N}(\boldsymbol{\mu}^+, \Sigma^+)$. Using Bayes rule, we have

$$\begin{aligned} p(\mathbf{x}|\mathbf{y}, \mathbf{z}) &\propto p(\mathbf{z}|\mathbf{y}, \mathbf{x}) p(\mathbf{x}|\mathbf{y}) \\ &= p(\mathbf{z}|\mathbf{x}) p(\mathbf{x}|\mathbf{y}), \end{aligned} \tag{3.68}$$

where the second equality is based on the fact that the two different sets of measurements $\mathbf{y}$ and $\mathbf{z}$ are conditionally independent given $\mathbf{x}$. Then, based on the derivation in Appendix A, we can get the updated estimates (posterior) of $\mathbf{x}$ as follows:

$$\boldsymbol{\mu}^+ = \boldsymbol{\mu}_{\mathbf{x}} + \Sigma_{\mathbf{x}} H^\top (H \Sigma_{\mathbf{x}} H^\top + R)^{-1} (\mathbf{z} - H \boldsymbol{\mu}_{\mathbf{x}}), \tag{3.69}$$

$$\Sigma^+ = \Sigma_{\mathbf{x}} - \Sigma_{\mathbf{x}} H^\top (H \Sigma_{\mathbf{x}} H^\top + R)^{-1} H \Sigma_{\mathbf{x}}. \tag{3.70}$$

Here $\boldsymbol{\mu}^+$ and $\Sigma^+$ are the MAP estimates. This is because $p(\mathbf{x}|\mathbf{z})$ is a Gaussian which achieves at its maximum its mean, and the MAP estimator finds the estimates which maximise the posterior distribution.

As (3.69) and (3.70) show, Bayesian fusion requires matrix inversion, which typically costs cubic time and square memory. They have a similar form to that of the update step in the EKF SLAM algorithm, as discussed in Section 2.1. This is because the Kalman filter is a particular case of the Bayes filter with an exact analytic solution due to it simplifying the system dynamics to be linear Gaussian (Mahler 2007).

### 3.5.2 Naïve Bayesian Fusion

*Naïve Bayesian fusion* refers to the situation when no cross-covariances are included, i.e. both $R$ and $\Sigma_{\mathbf{x}}$ in Section 3.5.1 are diagonal matrices. Therefore, Naïve Bayesian fusion is a point-to-point fusion method. For each point, given the prior $x_i \sim \mathcal{N}(\mu_x, \sigma_x^2)$ and the observation model

that $z_i = x_i + \delta_i$, $\delta_i \sim \mathcal{N}(0, \sigma_z^2)$, the posterior mean and variance are

$$(\sigma^+)^2 = \frac{1}{\frac{1}{\sigma_x^2} + \frac{1}{\sigma_z^2}} = \frac{\sigma_x^2 \sigma_z^2}{\sigma_x^2 + \sigma_z^2}, \tag{3.71}$$

$$\mu^+ = (\sigma^+)^2 \left( \frac{z_i}{\sigma_z^2} + \frac{\mu_x}{\sigma_x^2} \right) = \frac{\sigma_x^2 z_i}{\sigma_x^2 + \sigma_z^2} + \frac{\sigma_z^2 \mu_x}{\sigma_x^2 + \sigma_z^2}. \tag{3.72}$$

### 3.5.3 Information-form Bayesian Fusion

Bayesian fusion with canonical parameters, i.e. information vector and information matrix, and moment parameters, mean vector and covariance matrix, are functionally equivalent. They are dual to each other. However, as can be seen from Theorem 3.1 to 3.4, different parametrisation makes what is computationally complex in one to be simple in the other (and vice versa).

To explain the information-form Bayesian fusion, we first denote the canonical parameters corresponding to those in Section 3.5.1, to be

$$Q_{\mathbf{x}} = \Sigma_{\mathbf{x}}^{-1}, \quad \boldsymbol{\eta}_{\mathbf{x}} = \Sigma_{\mathbf{x}}^{-1} \boldsymbol{\mu}_{\mathbf{x}}, \quad Q_R = R^{-1}. \tag{3.73}$$

Then, by substituting the canonical parameters in (3.33) and (3.34), we can obtain the posterior with the updated information vector $\boldsymbol{\eta}^+$ and information matrix $Q^+$ as follows:

$$p(\mathbf{x}|\mathbf{y}, \mathbf{z}) = \mathcal{N}_c(\mathbf{x}|\boldsymbol{\eta}^+, Q^+), \quad \text{where} \tag{3.74}$$

$$Q^+ = Q_{\mathbf{x}} + H^\top Q_R H, \tag{3.75}$$

$$\boldsymbol{\eta}^+ = \boldsymbol{\eta}_{\mathbf{x}} + H^\top Q_R \mathbf{z}. \tag{3.76}$$

The derivation can be found in Appendix A.

Specifically, when $H$ is an identity matrix, (3.75) and (3.76) have even simpler forms, which are

$$Q^+ = Q_{\mathbf{x}} + Q_R, \tag{3.77}$$

$$\boldsymbol{\eta}^+ = \boldsymbol{\eta}_{\mathbf{x}} + Q_R \mathbf{z}. \tag{3.78}$$

As shown in (3.77), it only requires the addition operation to update the information matrix; while its dual form, as in (3.70), needs inverting the covariance matrix.

The mean and variance can be recovered from $\boldsymbol{\eta}^+$ and $Q^+$ using

$$\boldsymbol{\mu}^+ = (Q^+)^{-1}\boldsymbol{\eta}^+, \tag{3.79}$$

$$\Sigma^+ = (Q^+)^{-1}. \tag{3.80}$$

As shown in (3.79) and (3.80), mean and variance recovery are expensive for big data and dense $Q^+$. Therefore, the result of information-form Bayesian fusion, when running as constant time approach, may degrade as the map size increases. However, when $Q^+$ is sparse and has certain sparse structures, e.g. a band or a tridiagonal matrix, mean and variance recovery can be accelerated, as will later be discussed in Section 5.2.3 and Section 6.2.5.

## 3.6 GPBF: GP and Bayesian Fusion for 2.5D Mapping

GPBF is a fully correlated mapping approach which uses GP regression and correlated Bayesian fusion. The final result is an *global optimal map* as it includes all the self-correlations and cross-correlations amongst all measurements. The spatial correlation is learned using GP regression. The initial idea comes from Vidal-Calleja et al. (2013), which proposes to insert spatial correlation in thickness mapping within the standard Bayesian fusion, thus improving the estimation accuracy. By including spatial correlation, one point can update the estimated map values for all the neighbours.

### 3.6.1 Problem Statement

Given two datasets to be fused for building a 2.5D probabilistic map as an example. Define $\Psi_1 = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ contains $n$ point-referenced noisy measurements $y_i \in \mathcal{Y}$ taken at the 2D location $\mathbf{x}_i \in \mathcal{X}$. Assume the underlying noise-free quantity that is associated with $y_i$ is $\xi_i$, as in (3.38). The set $\mathcal{X}$ is a subset of $\mathbb{R}^2$, and $\mathcal{Y}$ is a subset of $\mathbb{R}$. Define $\Psi_2 = \left\{(\mathbf{x}_j^*, z_j)\right\}_{j=1}^{n^*}$, with $n^*$ sensor readings $z_j \in \mathcal{Z} \in \mathbb{R}$ measured from $\mathbf{x}_j^* \in \mathcal{X}^* \in \mathbb{R}^2$ locations. $\mathcal{X}$ and $\mathcal{X}^*$ are considered to be different. Assume measurements locations are known and accurate. Assume the measurements in $\Psi_1$ are at a lower resolution than those in $\Psi_2$. In the same regression problem as in (3.38), $\Psi_1$

is treated as the training set; and the training input $X$, the training output $\mathbf{y}$ and the query input $X^*$ are defined accordingly. Denote $\mathbf{z} = [z_1; z_2; \ldots; z_{n^*}]$.

### 3.6.2 Approach

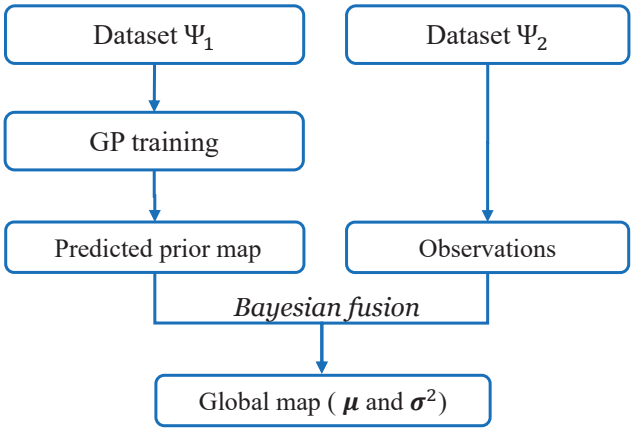



FIGURE 3.2: Flowchart of the GPBF framework.

As shown by the flowchart of GPBF in Figure 3.2, a GP is applied to learn from $\Psi_1$ to predict the high-resolution map $p(\boldsymbol{\xi}|X^*) = \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ at the test locations $X^*$, which is used as the prior map for fusion. In Bayesian fusion of these two datasets, define the likelihood $p(\mathbf{z}|\boldsymbol{\xi}, X^*)$ following that in Section 3.5.1. Then the prior map is updated with data from $\Psi_2$ by Bayesian fusion via MAP. Based on (3.69) and (3.70), the mean and covariance for the posterior map are given by

$$\boldsymbol{\mu}^+ = \boldsymbol{\mu} + \Sigma H^\top (H\Sigma H^\top + R)^{-1}(\mathbf{z} - H\boldsymbol{\mu}), \tag{3.81}$$

$$\Sigma^+ = \Sigma - \Sigma H^\top (H\Sigma H^\top + R)^{-1} H\Sigma. \tag{3.82}$$

GPBF is globally optimal in the sense that it includes all the auto- and cross-covariances of data points. Therefore, GPBF can cover the long-length correlations although with many fill-ins in the covariance matrix. These fill-ins, which represent the very weak correlation between faraway points, do not help much in improving the estimation accuracy while incurring much more computation than justified. Therefore, GPBF will soon become intractable as $X^*$ grows big.

## 3.7 Summary

This chapter has established the mathematical notation, preliminary and basic definitions and techniques, including MVNs, Bayes rule, GP and GMRF inference, that are applied throughout the thesis. The mapping problem that will be tackled in Chapter 4, 5 and 6 was stated in Section 3.6.1. In addition, the optimal global benchmark method, GPBF was introduced.

# Chapter 4

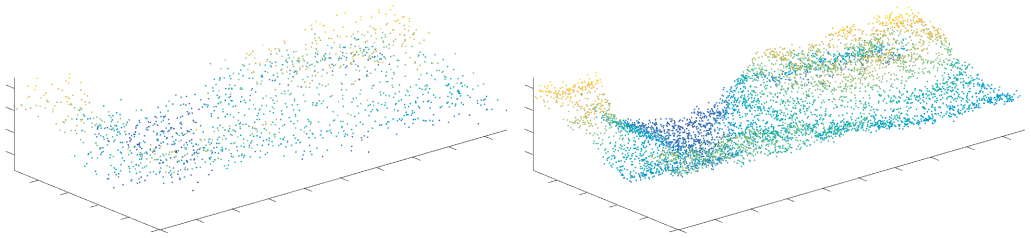
# Gaussian Processes for Bayesian Fusion with Conditionally Independent 2.5D Submapping

Traditional approaches for building large-scale 2.5D maps often assume that each cell in a map is independent on others. This assumption is not valid as real environments have an inherent structure. GPs is a straightforward and powerful tool for modelling data with spatial correlations. However, building large-scale 2.5D maps with GPs is confronted with prohibitive computation and memory consumption, as was explained in Section 3.3.2. To address this problem, a general probabilistic framework named **subGPBF** is developed, at which core lies an innovative CI submapping technique. Some of the principal ideas of subGPBF are:

- building CI submaps to address the scalability problem,
- fusing new incoming data incrementally while considering spatial correlations, and
- allowing information to be transmitted bi-directionally between CI submaps.

## 4.1 Problem Statement and Approach Overview

Given two or multiple sources of sensor data are observed, each containing noisy measurements covering the same 2D area. One dataset may comprise incomplete sparse measurements with higher uncertainty, e.g. Figure 4.1a; while another dataset is also incomplete but with lower uncertainty and denser sample locations, e.g. Figure 4.1b. To generate a high-resolution representation of the environment with lower uncertainty, it is often required to integrate multi-source data from homogeneous or heterogeneous sensors (see Section 2.3 and 2.5 for more details). We follow the problem statement in Section 3.6.1.



(A) $\Psi_1$ data points in 3D view (B) $\Psi_2$ data points in 3D view

FIGURE 4.1: Two synthetic terrain datasets. In 2.5D plots, elevation values are shown in colour, and the vertical axis corresponds to the latitude and horizontal axis is the longitude.

In subGPBF, GP regression is applied to learn the noise-free latent process from $\Psi_1$. Then the trained GP model can predict the prior map at the query locations $X^*$. When $X^*$ is large, we partition the full map into CI submaps, denoted as $s_1, s_2, \ldots, s_r$, where $r$ is the total number of submaps that cover the full map. The size of the submaps is chosen so as to minimise the correlation between every second submap. $X^*$ is divided into subsets, denoted as $\{X^*_{s_i}\}_{i=1}^{r}$, $X^*_{s_1:s_r} \in X^*$, where each $X^*_{s_i}$ contains the points that lie within $s_i$. Note that $X^*_{s_1} \cap X^*_{s_2} \neq 0$, *i.e.* consecutive submaps have an overlapping part. After the partitioning, the trained GP model is used to predict the prior submap, whose pdf is denoted as $s_i^- \sim p\left(\boldsymbol{\xi}_{s_i} | X^*_{s_i}\right)$. The correlated Bayesian fusion is then applied to fuse the observations from $\Psi_2$ with the prior submap $s_i^-$, or the *currently optimal*[1] submap $s_2^+, \ldots, s_r^+$ from the *forward update* process. The likelihood function $p(\mathbf{z}_{s_i} | \boldsymbol{\xi}_{s_i}, X^*_{s_i})$ is obtained via marginalisation from the full map $p(\mathbf{z}|\boldsymbol{\xi})$. After building all submaps, the *backward update* algorithm can propagate the influence of new information from the last submap backwards to all

[1] In this thesis, *currently optimal*, or *up-to-date*, means the estimates contain all the current information.

former submaps, thus generating the nearly optimal global map, i.e. mean and variances. Note that if the submaps are strictly conditionally independent, the global optimality of the algorithm is preserved with subGPBF. From now on we will consider that the CI assumption holds, although this might be an approximation in certain cases, for example, loop closure.

The flowchart of subGPBF is shown in Figure 4.2 and the implementation is in Algorithm 1. After the GP model is trained, CI submaps are built sequentially in the forward direction via prior mapping, correlated fusion and forward update; then, after creating all the submaps, one process of backward update will correct from the last to the first submap.

More intuitively, Figure 4.3 schematically shows how three submaps are built in the forward process. In general, Figure 4.4a shows the graphical model for all submaps; Figure 4.4b shows the elements of the full covariance matrix split into $r$ submaps and the information flow for the backward update. Figure 4.4b indicates that instead of building the dense, full covariance, subGPBF maintains a block tridiagonal matrix. Note that the overlapping area between consecutive submaps can be of arbitrary size.

## 4.2 SubGPBF: the Incremental CI Submapping Approach

### 4.2.1 The Graphical Model and CI Submaps

The relationship between submaps can be expressed either using a DGM or an Undirected Graphical Model (UGM). Unlike some other domains for which being forced to choose a direction for the edges is rather awkward, such as the GMs with circles, there is already a natural ordering of the sets of variables - submaps will be built one after the other. In the Bayes network model used in this thesis, as in Figure 4.4a, the nodes do not represent each variable but MVN vectors; and each submap contains two nodes. In Figure 4.4a, node $\boldsymbol{\xi}_j$ represents a set of components $j$ of the state at some locations on the map and $\mathbf{z}_j$ is the set of sensor measurements related to those areas. By partitioning the input domain based on their spatial locations, such Bayes network is ensured to be a Gaussian chain graph, with observations.

Without loss of generality, we will use a global map which contains two consecutive submaps for explanation. Let the global map contain three components $\boldsymbol{\xi}_a, \boldsymbol{\xi}_b, \boldsymbol{\xi}_c$. Define two MVN variables

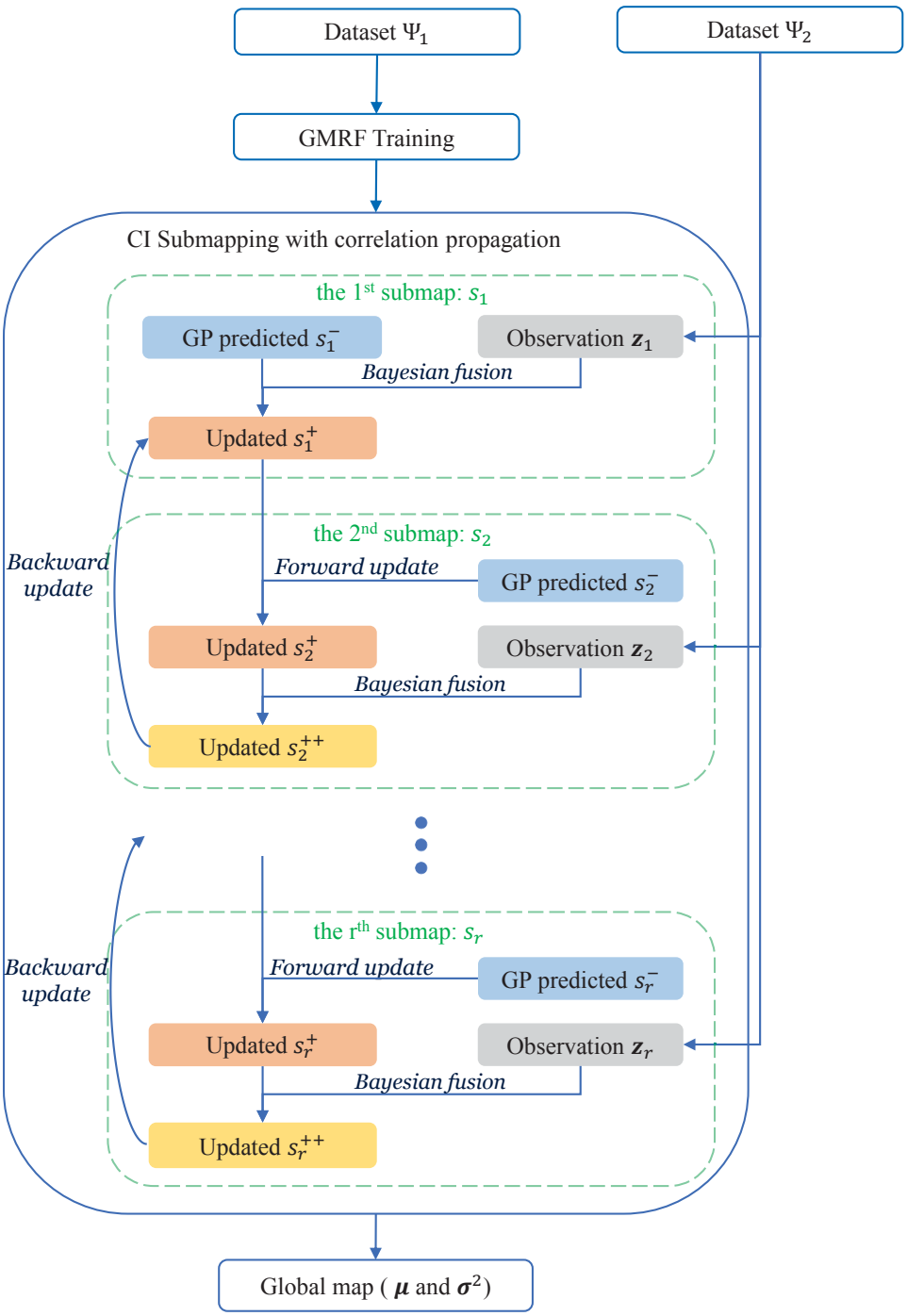



FIGURE 4.2: Flowchart of the proposed subGPBF framework.

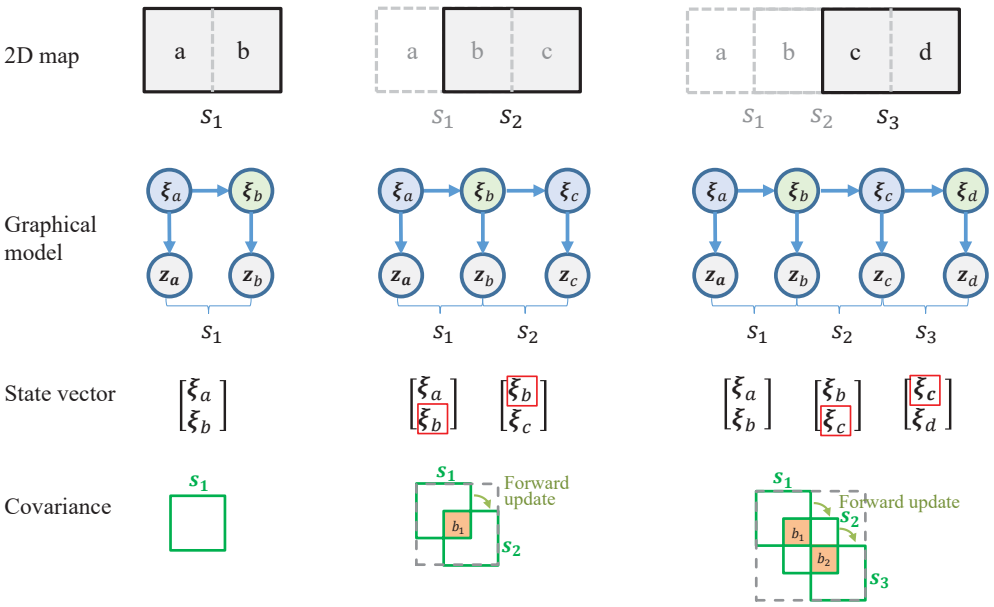



FIGURE 4.3: 2D demonstration of subGPBF, with an example of building three submaps. *The* $1^{st}$ *row*: 2D map location. *The* $2^{nd}$ *row*: the corresponding graphical model. *The* $3^{rd}$ *row*: the state vector of each submap to be estimated. $b_1, \ldots, b_{r-1}$ denote the shared elements between submaps. *The* $4^{th}$ *row*: the covariance matrix split into submaps.

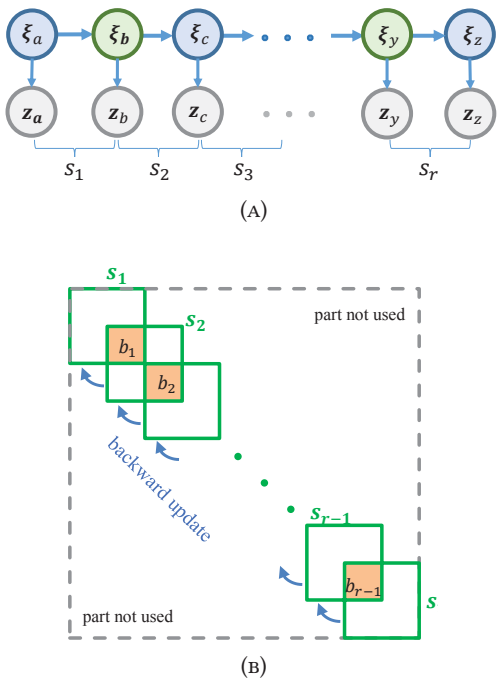



FIGURE 4.4: Schematic representation of all CI submaps. (A) A Bayes network showing the probabilistic dependencies between state $\boldsymbol{\xi}$ and measurements $\mathbf{z}$. (B) The full covariance matrix split into submaps in subGPBF, where $b_i$ represents the common elements shared between submaps.

**Algorithm 1** SubGPBF

**Require:**
Noisy dataset 1: $\Psi_1 = \{X, \mathbf{y}\}$
Noisy dataset 2: $\Psi_2 = \{X^*, \mathbf{z}\}$ and the partitioned subsets $\{X^*_{s_i}, \mathbf{z}_{s_i}\}_{i=1}^{r}$
**Ensure:** Optimal global map
**procedure** 1 GP Training
    Training a GP using $\Psi_1$
**end procedure**

**procedure** 2 Forward process:prior mapping, correlated fusion and forward update
    $s_1^- = GPprediction(X^*_{s_1})$
    $s_1^+ = correlatedFusion(s_1^-, \mathbf{z}_{s_1})$
    **for** $i = 2$ to $r$ **do**
        $s_i^- = GPprediction(X^*_{s_i})$
        $s_i^+ = forwardUpdate(s_{i-1}^+, s_i^-)$ ▷ Refer to Algorithm 2
        $\mathbf{z}_{s_i} = getObservations(\Psi_2)$
        $s_i^{++} = correlatedFusion(s_i^+, \mathbf{z}_{s_i})$ ▷ $s_i^{++}$ is currently optimal
    **end for**
    **return** $\{s_1^+, s_i^{++}\}$ where $i = 2, \ldots, r$
**end procedure**

**procedure** 3 Backward Update
    $s_r^{+++} \triangleq s_r^{++}$ ▷ Because $s_r^{++}$ is globally optimal given the CI condition is held
    **for** $i = r$ to 2 **do** ▷ For notation simplicity, $s_1^{++} \triangleq s_1^+$
        $s_{i-1}^{+++} = backwardUpdate(s_i^{+++}, s_{i-1}^{++})$ ▷ Refer to Algorithm 3
    **end for**
    **return** $\{s_i^{+++}\}$ where $i = 1, \ldots, r$ ▷ Globally optimal given CI condition is held
**end procedure**

$\boldsymbol{\xi}_{s_1}$ and $\boldsymbol{\xi}_{s_2}$, representing the two consecutive submaps, thus

$$\boldsymbol{\xi}_{s_1} = \begin{bmatrix} \boldsymbol{\xi}_a \\ \boldsymbol{\xi}_b \end{bmatrix}, \quad \boldsymbol{\xi}_{s_2} = \begin{bmatrix} \boldsymbol{\xi}_b \\ \boldsymbol{\xi}_c \end{bmatrix}. \tag{4.1}$$

As can be seen from Fig. 4.4a, the only connection between the set of nodes $(\boldsymbol{\xi}_a, \mathbf{z}_a)$ and $(\boldsymbol{\xi}_c, \mathbf{z}_c)$ is through node $\boldsymbol{\xi}_b$, i.e. subgraph $(\boldsymbol{\xi}_a, \mathbf{z}_a)$ and $(\boldsymbol{\xi}_c, \mathbf{z}_c)$ are *d-separated* given $\boldsymbol{\xi}_b$ (Koller and Friedman 2009). In other words, given $\boldsymbol{\xi}_b$, submaps $s_1$ and $s_2$ do not carry any additional information about each other. In Piniés and Tardós (2008), it is called the *submap CI property*, and submaps with such CI property are called CI submaps.

A well-known approximation in SLAM is to ignore the correlation between submaps that do not overlap. This thesis follows this approximation and chooses submaps that overlap to recover the

correlations. The submaps are by construction overlapping and therefore conditional independent. CI submaps can be seen as a *trade-off* between the fully correlated global map and the independent submaps. Information can be passed between submaps that allows a distributed and decentralised solution to be maintained, which is equivalent to a global map. In addition, the map partitioning resulting from the conditionally independent formulation has the potential for a reduction in the computational complexity since each submap is of smaller dimension than a single map. Please refer to Section 3.2.3 and 3.2.4 for the related knowledge about the CI property and Bayes network model.

The following sections will explain how to compute the optimal global map from CI submaps.

### 4.2.2 GP for Prior Mapping

The underlying noise-free process is learned from dataset $\Psi_1$ using GP regression, and then this GP is used to predict the prior map at the desired resolution, as was described in Section 3.3.2. The output is a correlated probabilistic map, including mean and covariance. When predicting at the query points subset $X^*_{s_1}$, the GP generates the prior submaps $s_1^-$, which is defined as

$$s_1^- \sim p(\boldsymbol{\xi}_{s_1}) = \mathcal{N}(\boldsymbol{\mu}_{s_1}, P_{s_1}), \tag{4.2}$$

where for the sake of notation simplicity, the query points locations $X^*_{s_1}$ and the training dataset $\Psi_1$ are ignored. Thus $p(\boldsymbol{\xi}_{s_1})$ is in short of $p(\boldsymbol{\xi}_{s_1}|X^*_{s_1}, \Psi_1)$. Using GP prediction allows us to increase or decrease map resolution (inferring more or fewer points) as required. Additionally, spatial correlations are encoded in the covariance matrix.

### 4.2.3 Spatially Correlated Bayesian Fusion

The likelihood function, denoted as $p(\mathbf{z}|\boldsymbol{\xi})$, is obtained in the same way as in Section 3.5.1 and 3.6.2. For $s_1$, the likelihood function $p(\mathbf{z}_a|\boldsymbol{\xi}_{s_1})$ is obtained via marginalisation from $p(\mathbf{z}|\boldsymbol{\xi})$, where $\mathbf{z}_a$ is the current observation for $s_1$.

With the prior submap $s_1^-$ and the likelihood function $p(\mathbf{z}_a|\boldsymbol{\xi}_{s_1})$, Bayesian fusion is performed to update $s_1^-$ with new information, and MAP estimation is applied to obtain the posterior submap

$s_1^+$. Define the pdf of $s_1^+$ to be $p(\boldsymbol{\xi}_{s_1}|\mathbf{z}_a) = \mathcal{N}(\boldsymbol{\mu}_{s_1}^a, P_{s_1}^a)$. The mean $\boldsymbol{\mu}_{s_1}^a$ and covariance $P_{s_1}^a$ are computed using (3.69) and (3.70), respectively. For MVNs, $\boldsymbol{\mu}_{s_1}^a$ and $P_{s_1}^a$ can be partitioned as

$$s_1^+ \sim p(\boldsymbol{\xi}_{s_1}|\mathbf{z}_a) = \mathcal{N}(\boldsymbol{\mu}_{s_1}^a, P_{s_1}^a) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_a^a \\ \boldsymbol{\mu}_b^a \end{bmatrix}, \begin{bmatrix} P_a^a & P_{ab}^a \\ P_{ba}^a & P_b^a \end{bmatrix}\right), \tag{4.3}$$

where the superscript indicates the set of new observations from $\Psi_2$ that have been incorporated into the estimation of the submap, and the subscript denotes the set of variables to be estimated.

The Bayesian fusion approach used here is referred to as the *correlated Bayesian fusion*, to distinguish it from the naïve Bayesian fusion described in Section 3.5.2. As noted above, the correlated Bayesian fusion includes spatial correlations in $P_{s_1}$, which is produced by a GP. As a result, one single measurement could update all its neighbours in the correlated Bayesian fusion, while in naïve Bayesian fusion it only updates the corresponding point value since the diagonal matrix $P_{s_1}$ only contains the uncorrelated variances.

Note that in the general case of $r$ submaps, except the last submap $s_r$ that is updated with both the common and non-common observations, the other submaps $s_i$, $i = 1, \ldots, r-1$ are only fused with the non-common observations. Another thing to notice is that from the second submap $s_2$ to $s_r$, observations from $\Psi_2$ are fused with the output of the forward update algorithm, not the output of GP prediction. For example, instead of building $s_2$ from scratch, the forward update algorithm, as will be described in Section 4.2.4, is applied to transmit information from $s_1^+$ to the GP predicted $s_2^-$, thus producing the currently optimal submap $s_2^+$. Then, $s_2^+$ is fused with $\mathbf{z}_b$ and $\mathbf{z}_c$, thus generating $s_2^{++}$. The detailed procedures can be found in Figure 4.2 and Algorithm 1.

### 4.2.4 Forward Update

The prior estimate of the second submap, denoted as $s_2^-$, can be computed by GP inference, and we denote it as

$$s_2^- \sim p(\boldsymbol{\xi}_{s_2}) = \mathcal{N}\left(\boldsymbol{\mu}_{s_2}, P_{s_2}\right) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_b \\ \boldsymbol{\mu}_c \end{bmatrix}, \begin{bmatrix} P_b & P_{bc} \\ P_{cb} & P_c \end{bmatrix}\right). \tag{4.4}$$

The goal is to compute the currently optimal submap 2, denoted as $s_2^+$, which contains the current observation $\mathbf{z}_a$. Define $s_2^+$ as

$$s_2^+ \sim p(\xi_{s_2}|\mathbf{z}_a) = \mathcal{N}(\boldsymbol{\mu}_{s_2}^a, P_{s_2}^a). \tag{4.5}$$

Define the pdf of the currently optimal local map[2] as

$$p(\xi_a, \xi_b, \xi_c|\mathbf{z}_a) \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_a^a \\ \boldsymbol{\mu}_b^a \\ \boldsymbol{\mu}_c^a \end{bmatrix}, \begin{bmatrix} P_a^a & P_{ab}^a & P_{ac}^a \\ P_{ba}^a & P_b^a & P_{bc}^a \\ P_{ca}^a & P_{cb}^a & P_c^a \end{bmatrix}\right). \tag{4.6}$$

Since $s_1^+$ has been updated with $\mathbf{z}_a$ while $s_2^-$ has not, $s_1^+$ in (4.3) coincides exactly with the first two blocks of the posterior map in (4.6) based on the marginalisation property of MVNs. Therefore, to get the currently optimal map, only the terms related to $\xi_c$ need to be computed, which requires information to be propagated forwards.

Before proposing the forward update algorithm, let us recall the submap CI property. It is observed from Figure 4.4a that the set of nodes $(\xi_a, \mathbf{z}_a)$ and $\xi_c$ are d-separated given $\xi_b$. This in turn implies that the components of $(\xi_a, \mathbf{z}_a)$ and $\xi_c$ are conditionally independent given $\xi_b$, namely

$$(\xi_a, \mathbf{z}_a) \perp \xi_c | \xi_b \,. \tag{4.7}$$

Then based on this submap CI property, we have the following proposition.

**Proposition 4.1.** *The currently optimal map, described by $p(\xi_a, \xi_b, \xi_c|\mathbf{z}_a)$, can be recovered from the currently optimal submap $s_1^+ \sim p(\xi_a, \xi_b|\mathbf{z}_a)$ and the submap $s_2^- \sim p(\xi_b, \xi_c)$.*

*Proof.* Based on the chain rule and (4.7), we get

$$\begin{aligned} p(\xi_a, \xi_b, \xi_c|\mathbf{z}_a) &= p(\xi_c|\xi_a, \xi_b, \mathbf{z}_a) p(\xi_a, \xi_b|\mathbf{z}_a) \\ &= p(\xi_c|\xi_b) p(\xi_a, \xi_b|\mathbf{z}_a), \end{aligned} \tag{4.8}$$

where the second equality comes from the CI property. We can observe in (4.8) that the first factor comes from submap $s_2^-$ (see (4.4)) by conditioning; and the second factor comes from submap $s_1^+$

[2] In this thesis, a *local map* refers to a map than contains two consecutive submaps. In this example, the local map is equal to the global map since only two submaps are considered.

(see (4.3)). Therefore, all information needed to recover the optimal map can be obtained from the information stored in each submap. □

Notice that in Proposition 4.1, no assumptions have been made about the particular distribution of the probability densities. Moreover, the factorisation only depends on general probabilistic theorems and the CI assumption.

Then we propose the forward update algorithm as follows.

**Corollary 4.2** (Forward update)**.** *After submap $s_1^+$ and $s_2^-$ are built, the currently optimal submap $s_2^+$ can be obtained using Algorithm 2. The mean and covariance of $s_2^+$ are obtained as*

$$\boldsymbol{\mu}_{s_2}^a = \begin{bmatrix} \boldsymbol{\mu}_b^a \\ \boldsymbol{\mu}_c + P_{cb}P_b^{-1}(\boldsymbol{\mu}_b^a - \boldsymbol{\mu}_b) \end{bmatrix}, \tag{4.9}$$

$$P_{s_2}^a = \begin{bmatrix} P_b^a & P_b^a P_b^{-1} P_{bc} \\ P_{cb}P_b^{-1}P_b^a & P_c + P_{cb}P_b^{-1}(P_b^a P_b^{-1} P_{bc} - P_{bc}) \end{bmatrix}. \tag{4.10}$$

*Besides, the cross-correlation term $P_{ac}^a$ in the currently optimal local map (see* (4.6)*) can also be recovered.*

**Algorithm 2** Forward update

1: $K_f \triangleq P_{cb}P_b^{-1}$;
2: $P_{cb}^a = K_f P_b^a$;
3: $P_c^a = P_c + K_f(P_{bc}^a - P_{bc})$;
4: $\boldsymbol{\mu}_c^a = \boldsymbol{\mu}_c + K_f(\boldsymbol{\mu}_b^a - \boldsymbol{\mu}_b)$;
5: (optionally) $P_{ac}^a = P_{ab}^a P_b^{-1} P_{bc}$.

*Proof.* See Appendix B. An alternative proof is in Appendix C, which follows Piniés and Tardós (2008).

Algorithm 2 shows that the forward update algorithm takes advantage of the available estimation of the state vectors that are in the border between both submaps. It also shows that all the terms related with $\boldsymbol{\xi}_c$ can be computed, thus the joint density in (4.6) can be recovered if needed. More importantly, the forward update algorithm allows submap 2 to add the observation $\mathbf{z}_a$ without referring to the full map $p(\boldsymbol{\xi}_a, \boldsymbol{\xi}_b, \boldsymbol{\xi}_c|\mathbf{z}_a)$. In fact, there is no need to recover the missing cross-correlation between submaps, $P_{ac}^a$, since it is not needed to get the currently optimal mean and

variances; yet $P^a_{ac} = P^a_{ab} P_b^{-1} P_{bc}$ can be computed if required. Therefore we can always build currently optimal submaps using the forward update algorithm one after the other, without having to compute the correlations between submaps.

Having calculated $s_2^+$ using the forward update algorithm, we then update $s_2^+$ with the current new observations $[\mathbf{z}_b; \mathbf{z}_c]$ via the correlated Bayesian fusion. Since $s_2$ is the last submap in the example, both the common and non-common observations are incorporated. Define $s_2^{++}$, which is globally optimal, as

$$s_2^{++} \sim p(\boldsymbol{\xi}_{s_2} | \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c) = \mathcal{N}(\boldsymbol{\mu}^{abc}_{s_2}, P^{abc}_{s_2}) = \mathcal{N}\left( \begin{bmatrix} \boldsymbol{\mu}^{abc}_b \\ \boldsymbol{\mu}^{abc}_c \end{bmatrix}, \begin{bmatrix} P^{abc}_b & P^{abc}_{bc} \\ P^{abc}_{cb} & P^{abc}_c \end{bmatrix} \right). \tag{4.11}$$

For more than two submaps, Algorithm 2 can be adopted to get the initial state estimate of the third submap $s_3^+$ using $s_2^{++}$ (see (4.11)) and the GP predicted $s_3^-$; then the correlated Bayesian fusion can be applied to update $s_3^+$. In such a sequential manner, submaps are built consecutively in the forward direction. After creating all submaps in the forward direction, by using GP prediction, correlated Bayesian fusion and the forward update algorithm, the last submap $s_r$ is globally optimal. This is because the last submap, denoted as $s_r^{++} \sim p(\boldsymbol{\xi}_{s_r} | \mathbf{z}_{s_1}, \ldots, \mathbf{z}_{s_r})$, comprises all the information from the former submaps. However, the other submaps $s_1, \ldots, s_{r-1}$ still do not contain information from the later submaps or more recent observations. Nonetheless, the optimal global map can be obtained via the backward update algorithm.

### 4.2.5 Backward Update

Having built the two CI submaps $s_1^+ \sim p(\boldsymbol{\xi}_a, \boldsymbol{\xi}_b | \mathbf{z}_a)$ (see (4.3)) and $s_2^{++} \sim p(\boldsymbol{\xi}_b, \boldsymbol{\xi}_c | \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c)$ (see (4.11)), we aim to compute the globally optimal submap 1, denoted as $s_1^{++}$, and optionally the joint distribution of the two submaps, both of which contain all the current observations $[\mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c]$. The globally optimal submap $s_1^{++}$ is defined as

$$s_1^{++} \sim p(\boldsymbol{\xi}_a, \boldsymbol{\xi}_b | \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c) = \mathcal{N}(\boldsymbol{\mu}^{abc}_{s_1}, P^{abc}_{s_1}). \tag{4.12}$$

Define the globally optimal local map to be

$$p(\boldsymbol{\xi}_a, \boldsymbol{\xi}_b, \boldsymbol{\xi}_c | \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c) = \mathcal{N}\left( \begin{bmatrix} \boldsymbol{\mu}_a^{abc} \\ \boldsymbol{\mu}_b^{abc} \\ \boldsymbol{\mu}_c^{abc} \end{bmatrix}, \begin{bmatrix} P_a^{abc} & P_{ab}^{abc} & P_{ac}^{abc} \\ P_{ba}^{abc} & P_b^{abc} & P_{bc}^{abc} \\ P_{ca}^{abc} & P_{cb}^{abc} & P_c^{abc} \end{bmatrix} \right). \tag{4.13}$$

As was discussed in Section 4.2.4, to compute the joint density in (4.13), only the terms related to $\boldsymbol{\xi}_a$ need to be computed. Therefore, information needs to be propagated backwards.

Given the submap CI property that

$$(\boldsymbol{\xi}_a, \mathbf{z}_a) \perp (\boldsymbol{\xi}_c, \mathbf{z}_b, \mathbf{z}_c) | \boldsymbol{\xi}_b \,, \tag{4.14}$$

we propose the following proposition.

**Proposition 4.3.** *The globally optimal map, described by $p(\boldsymbol{\xi}_a, \boldsymbol{\xi}_b, \boldsymbol{\xi}_c | \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c)$, can be recovered from the globally optimal submap $s_2^{++} \sim p(\boldsymbol{\xi}_b, \boldsymbol{\xi}_c | \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c)$ and the submap $s_1^{+} \sim p(\boldsymbol{\xi}_a, \boldsymbol{\xi}_b | \mathbf{z}_a)$.*

*Proof.* Based on the chain rule and the submap CI property in (4.14), we get

$$\begin{aligned} p(\boldsymbol{\xi}_a, \boldsymbol{\xi}_b, \boldsymbol{\xi}_c | \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c) &= p(\boldsymbol{\xi}_a | \boldsymbol{\xi}_b, \boldsymbol{\xi}_c, \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c) p(\boldsymbol{\xi}_b, \boldsymbol{\xi}_c | \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c) \\ &= p(\boldsymbol{\xi}_a | \boldsymbol{\xi}_b, \mathbf{z}_a) p(\boldsymbol{\xi}_b, \boldsymbol{\xi}_c | \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c) \,, \end{aligned} \tag{4.15}$$

where the second equality comes from (4.14). In (4.15), observe that the first factor comes from $s_1^{+}$ in (4.3) by conditioning; and the second factor comes from $s_2^{++}$ in (4.11). Therefore, the globally optimal map can be recovered from the information stored in the two submaps. □

Similar to Proposition 4.1, Proposition 4.3 does not depend on the particular distribution of the probability densities, but is based only on general probabilistic theorems and the CI assumption.

Then we propose the backward update algorithm in Corollary 4.4.

**Corollary 4.4** (Backward update). *After submap $s_1^+$ and $s_2^{++}$ are built, the globally optimal submap $s_1^{++}$ can be obtained using Algorithm 3*

$$\boldsymbol{\mu}_{s_1}^{abc} = \begin{bmatrix} \boldsymbol{\mu}_a^a + P_{ab}^a (P_b^a)^{-1} (\boldsymbol{\mu}_b^{abc} - \boldsymbol{\mu}_b^a) \\ \boldsymbol{\mu}_b^{abc} \end{bmatrix}, \tag{4.16}$$

$$P_{s_1}^{abc} = \begin{bmatrix} P_a^a + P_{ab}^a (P_b^a)^{-1} (P_b^{abc} (P_b^a)^{-1} P_{ba}^a - P_{ba}^a) & P_{ab}^a (P_b^a)^{-1} P_b^{abc} \\ P_b^{abc} (P_b^a)^{-1} P_{ba}^a & P_b^{abc} \end{bmatrix}. \tag{4.17}$$

*Besides, the cross-correlation term $P_{ac}^{abc}$ in the globally optimal local map (see* (4.13)*) can also be recovered.*

**Algorithm 3** Backward update

1: $K_b \triangleq P_{ab}^a (P_b^a)^{-1}$;
2: $P_{ab}^{abc} = K_b P_b^{abc}$;
3: $P_a^{abc} = P_a^a + K_b (P_{ba}^{abc} - P_{ba}^a)$;
4: $\boldsymbol{\mu}_a^{abc} = \boldsymbol{\mu}_a^a + K_b (\boldsymbol{\mu}_b^{abc} - \boldsymbol{\mu}_b^a)$;
5: (optionally) $P_{ac}^{abc} = K_b P_{bc}^{abc}$.

*Proof.* See Appendix D.

Corollary 4.4 shows that backward update algorithm is also a data sharing technique as is the forward update algorithm. It uses the difference in the estimation of the shared parts between two submaps to update the previous submap. It also shows that all the terms related with $\boldsymbol{\xi}_a$ can be computed, thus the full map in (4.13) can be recovered if needed. The backward update algorithm allows submap 1 to add all the current observations $[\mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c]$ without referring to the full map in (4.13). In fact, there is no need to recover the missing cross-correlation between submaps, e.g. $P_{ac}^{abc}$, although it could potentially be recovered to get the full map if needed.

Notice that the forward update algorithm and the backward update algorithm can be used separately or together. Here they are used together to ensure the optimality of subGPBF. By now, the globally optimal mean i.e. $\boldsymbol{\mu}_a^{abc}$, $\boldsymbol{\mu}_b^{abc}$, $\boldsymbol{\mu}_c^{abc}$, and the variances, i.e. the diagonal entries of $P_a^{abc}$, $P_b^{abc}$, $P_c^{abc}$, have been obtained.

## 4.3 Comparison

### 4.3.1 Forward Update and Backward Update

When the new observations come in, the estimation of the common part between two nearby submaps will change. On account of this fact, both the forward and backward update algorithms use the estimation difference in mutual components to correct the state, enforcing the cross-correlation forwards or backwards to other submaps. The deduction of the two algorithms is based on the CI assumption and general probability theorems.

The forward and backward update algorithms are used for different goals in subGPBF. The forward update algorithm initialises the following submap to contain all the current observations, thus being currently optimal. In addition, the forward update algorithm is used together with Bayesian fusion, GP prediction to build submaps incrementally. The backward update algorithm corrects from the last to the first submap in a single process, and finally generates the optimal global mean and variances of the variables to be estimated.

### 4.3.2 Forward Update and Augmentation

The forward update algorithm resembles the Augmentation algorithm in EKF SLAM (see Section 2.1), in the sense that the previous submap is used to build the following submap. One of the differences is that EKF SLAM requires a known state transition model to propagate from the current state and further augment the state using newly observed landmarks. The transition model is required in EKF SLAM to predict the mean and covariance of the state in future steps. In the forward update algorithm, locations are given, and it is just assumed that the previous and following submap are next to each other in space. Therefore forward update algorithm does not require a known transition model.

## 4.4 Variants and Applications

SubGPBF is flexible, and with some minor changes, it can be widely applicable to different problems. For instance, Section 4.4.1 shows the forward update algorithm can be used for prediction,

and Section 4.4.2 illustrates how to adapt subGPBF for sequential mapping with one source of big data. In addition, subGPBF can be readily used for mapping with multiple datasets, when the last fusion result is regarded as the prior and the mapping process is repeated.

### 4.4.1 Adapting the Forward Update Algorithm to Prediction

This subsection illustrates that the forward update algorithm can be used to augment the state vector, with function values inferred at a finite set of new input points at arbitrary locations.

Let $f(\mathbf{x})$ be an unknown non-linear function over the input vector space $\mathbf{x}$. Assume we obtain some observations $\mathbf{y}$ at some inputs $X$. We use the same functional dependency model as in (3.38), and assume the noise is iid Gaussian with the variance of $\sigma^2$. Suppose we wish to infer the function values at a set of inputs $X^*$. Under MVN distribution, given the prior joint probability over $f(X)$ and $f(X^*)$, the function values at inputs $X^*$ can be inferred using the forward update algorithm, as described below.

Define the prior joint distribution over $\mathbf{y}$ and $f(X^*)$ to be

$$\begin{bmatrix} \mathbf{y} \\ f_0(X^*) \end{bmatrix} \sim \mathcal{N}\left( \begin{bmatrix} \boldsymbol{\mu}_X \\ \boldsymbol{\mu}_{X^*} \end{bmatrix}, \begin{bmatrix} P_X + \sigma^2 I & P_{X,X^*} \\ P_{X^*,X} & P_{X^*} \end{bmatrix} \right). \tag{4.18}$$

Suppose $\mathbf{y}$ is updated with some new incoming information, and denote the posterior of $\mathbf{y}$ as

$$\mathbf{y}^+ \sim \mathcal{N}(\boldsymbol{\mu}_X^+, P_X^+), \tag{4.19}$$

Once assuming the new information that updates $\mathbf{y}$ is conditionally independent of $f(X^*)$ given $f(X)$, we can directly apply Algorithm 2 to get the posterior of $f(X^*)$ as follows

$$f^+(X^*) \sim \mathcal{N}(\boldsymbol{\mu}_{X^*}^+, P_{X^*}^+), \quad \text{where} \tag{4.20}$$

$$\boldsymbol{\mu}_{X^*}^+ = \boldsymbol{\mu}_{X^*} + P_{X^*,X}(P_X + \sigma^2 I)^{-1}(\boldsymbol{\mu}_X^+ - \boldsymbol{\mu}_X), \tag{4.21}$$

$$P_{X^*}^+ = P_{X^*} + P_{X^*,X}(P_X + \sigma^2 I)^{-1}((P_X^+)^{-1} P_X^{-1} P_{X,X^*} - P_{X,X^*}). \tag{4.22}$$

On the other hand, the problem in this section can be thought of as a special case in (4.1) that

$$\xi_{s_1} = \begin{bmatrix} \mathbf{y} \end{bmatrix}, \quad \xi_{s_2} = \begin{bmatrix} \mathbf{y} \\ f(X^*) \end{bmatrix}. \tag{4.23}$$

Furthermore, the results deduced above in turn indicate that the size of the overlapping part between two consecutive CI submaps can be of any size, as was mentioned in Section 4.1.

Both the posterior mean (see (4.21)) and covariance ((4.22)) for any arbitrary individual query points $X^*$ can be determined. Thus, the forward update algorithm allows us to interpolate values at new inputs without the need to refer to the information that updated $\mathbf{y}$.

### 4.4.2 Adapting SubGPBF to Sequential Mapping with One Data Source

SubGPBF can be adapted to building a large-scale map with one source of sensor data. It is suitable for the online incremental submapping task. The whole process is shown in Algorithm 4. The mathematical proof and deduction are in Appendix H.

The main difference with subGPBF is that backward update algorithm alone can generate the optimal global map. The forward update is not needed as the latest submap is already optimal; therefore only GP inference, Bayesian fusion and the backward update are required. In addition, to account for the arbitrarily big amount of sensor data, each local GP is trained for each submap and optimisation of the hyperparameters is within each submap. To further save training time and better scale for real-time scenario, the learned hyperparameters of the former submap are used as the initial value in optimisation of the subsequent submap. Such hyperparameter learning methods can also be used in subGPBF; although they are not used because we have assumed $\Psi_1$ has a limited amount of data.

## 4.5 Summary

This chapter proposed subGPBF - a general, probabilistic framework for large-scale mapping with multiple sources of data. The main techniques used were GP regression, spatially correlated Bayesian fusion, and the CI submapping strategies. The spatial correlations, learned via GP

**Algorithm 4** SubGPBF for one dataset

**Require:**
Training dataset subsets: $\{X_{s_i}, \mathbf{y}_{s_i}\}_{i=1}^{r}$
Query points subsets: $X^*_{s_i}$, $i = 1, \ldots, r$
**Ensure:** Optimal global map
**procedure** 1 Forward mapping:prior mapping and correlated fusion
    $LGP_1 = GPtraining(X_{s_1}, \mathbf{y}_{s_1})$ ▷ The local GP
    $s_1^- = GPprediction(X^*_{s_1})$
    **for** $i = 2$ to $r$ **do**
        $LGP_i$=$GPtraining(X_{s_i}, \mathbf{y}_{s_i})$ ▷ Hyperparameters of $s_i$ are initialised using that of $s_{i-1}$.
        $s_i^- = GPprediction(X^*_{s_i})$
        $\mathbf{z}_{s_i} = getObservations$ ▷ $\mathbf{z}_{s_i} = marginalisation(s_{i-1})$
        $s_i^+ = correlatedFusion(s_i^-, \mathbf{z}_{s_i})$
    **end for**
    **return** $\{s_1^-, s_i^+\}$ where $i = 1, \ldots, r$
**end procedure**

**procedure** 2 Backward Update
    $s_r^{++} \triangleq s_r^+$ ▷ Because $s_r^+$ is already globally optimal
    **for** $i = r - 1$ to 2 **do** ▷ For notation simplicity, $s_1^+ \triangleq s_1^-$
        $s_i^{++} = backwardUpdate(s_{i+1}^{++}, s_i^+)$
    **end for**
    $s_1^{++} = backwardUpdate(s_2^{++}, s_1^-)$
    **return** $\{s_i^{++}\}$ where $i = 1, \ldots, r$ ▷ Globally optimal given CI condition is held
**end procedure**

regression, were incorporated into Bayesian fusion to improve accuracy and consistency. The CI submapping strategies were developed to address the scalability problem when using GP and correlated Bayesian fusion for big data. The innovation lies in the two novel data sharing techniques, the forward and backward update algorithms. By using them together, information can be transmitted bi-directionally between CI submaps. Finally, after using the backward update algorithm to correct all previous submaps, subGPBF is able to obtain the optimal global map, with no other assumptions except the CI property of submaps.

# Chapter 5

# Gaussian Markov Random Fields for Bayesian Fusion and 2.5D Mapping

While Chapter 4 used the submapping techniques to approximate the global mapping method using GP and Bayesian fusion, this chapter develops a global approximation approach in the information form. The main advantage of moving from the covariance form to the information form stems from the better computational performance. The motivations are in two aspects:

- GMRF-SPDEs closely approximate the Matérn GRFs regarding computational efficiency and accuracy (Simpson et al. 2012; Bolin and Lindgren 2013), and
- the correlated Bayesian fusion is more efficient in the information form than in its dual.

Therefore the proposed approach is named as **GMRF-BF**. GMRF-SPDEs, as was described in Section 3.4.2, are applied in this chapter and Chapter 6 to model the spatially correlated data and infer high-resolution maps. The information-form Bayesian fusion, as was introduced in Section 3.5.3, is then used to update the prior map predicted by the GMRF-SPDE.

The contributions are twofold. Although neither the continuously indexed GMRF-SPDE nor the information-form Bayesian fusion is new, GMRF-BF firstly combines these two methods to obtain a further computational gain when mapping in 2D. In addition, the direct link between

the hyperparameters of a GMRF-SPDE and those of the information matrix $Q$ has been found, and the details are explained in Appendix E. This link is used to build the information matrix.

## 5.1 Problem Statement and Approach Overview

Following the problem statement and dataset definition in Section 4.1, GMRF-BF is developed in consideration of the computational gain achieved by combining the GMRF representation and the information-form Bayesian fusion. Firstly, a GMRF-SPDE is applied to learn the noise-free latent process from dataset $\Psi_1$, and then to predict the information-form *prior map*, denoted as $p(\boldsymbol{\xi}|X^*)$. When dataset $\Psi_2$ comes, the likelihood function $p(\mathbf{z}|\boldsymbol{\xi}, X^*)$ can be obtained when a linear Gaussian observation model is used. Then the information-form Bayesian fusion is used to integrate the prior map with the observation likelihood to get the posterior map, which will be in the information form. The MAP estimator is applied to get the posterior distribution $p(\boldsymbol{\xi}|\mathbf{z}, X^*) \propto p(\boldsymbol{\xi}|X^*)p(\mathbf{z}|\boldsymbol{\xi}, X^*)$. Finally, the mean and uncertainty maps are recovered.

The flowchart of GMRF-BF is shown in Figure 5.1. As the flowchart shows, GMRF-BF has a comparable and parallel structure with GPBF (see the flowchart in Figure 3.2). However, GMRF-BF adapts all computations , except the map recovery, in the information form.

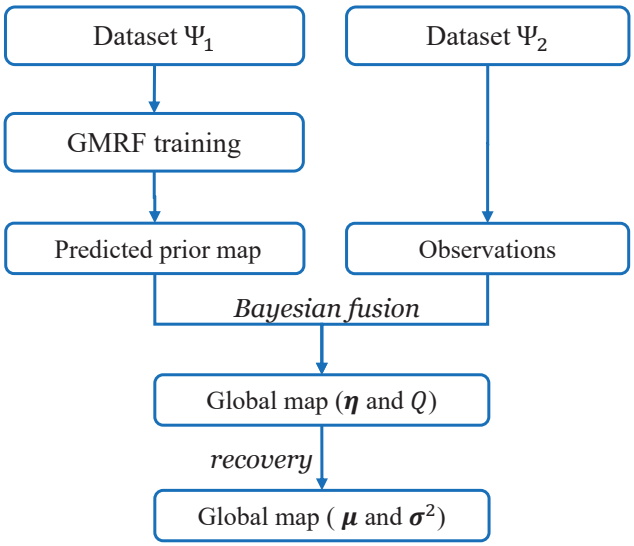



Figure 5.1: Flowchart of the proposed GMRF-BF framework.

## 5.2 GMRF-BF: the Information-form Global Mapping Approach

### 5.2.1 GMRF-SPDE for Prior Mapping

GMRF-SPDEs explicitly map the continuous latent field, thus no information will be lost due to binning data locations to regular grid cells. In addition, GMRF-SPDEs can learn spatial correlations and make inference concerning the correlations. Furthermore, a GMRF-SPDE is defined by a sparse matrix, which allows for computationally effective numerical methods.

Here we consider the same regression problem as in Section 3.4.2 and use $\Psi_1$ as the training dataset. The noisy data are modelled using a zero-mean GP with the Matérn kernel, which is the same as in subGPBF, yet the computation is done using the corresponding GMRF model. When a zero mean function is used, the GMRF-SPDE model is specified by the hyperparameters in the Matérn covariance function. The only hyperparameter that is chosen manually, in a similar way as for the GPs, is the smoothness parameter $\nu$, which decides the number of neighbours for each single point. Then the other hyperparameters of the Matérn kernel and the data noise variance $\sigma_\epsilon^2$ are learned by solving the SPDE using finite element methods. This approach has been proven to be the best linear approximation to the continuous solution to the SPDE by Lindgren et al. (2011). Then the learned hyperparameters are used to build the spatially correlated prior map, which is

$$p(\boldsymbol{\xi}) = \mathcal{N}_c(\boldsymbol{\eta}, Q), \quad \text{where} \tag{5.1}$$

$$\boldsymbol{\eta} = Q(X^*, X)\mathbf{y}, \tag{5.2}$$

$$Q = Q(X^*, X^*) + Q_\epsilon. \tag{5.3}$$

For the sake of notation simplicity, the query points locations $X^*$ and the training dataset $\Psi_1$ are ignored. Therefore $p(\boldsymbol{\xi})$ in (5.1) is short for $p(\boldsymbol{\xi}|X^*, \Psi_1)$. The information matrices $Q(X^*, X)$ and $Q(X^*, X^*)$ are calculated using the approach described in Appendix E, and note that in this case they are sparse. $\mathbf{y}$ denotes the observations in $\Psi_1$, $Q_\epsilon = \sigma_\epsilon^{-2} I_n$ is a diagonal matrix that represents the precision of data itself, i.e. the inverse of variances of data noise.

The mean state vector that corresponds to $\boldsymbol{\eta}$ in (5.2) is

$$\boldsymbol{\mu} = Q^{-1} Q(X^*, X)\mathbf{y}. \tag{5.4}$$

The comparison between (5.2) and (5.4) shows that maintaining $\boldsymbol{\eta}$ is cheaper than computing $\boldsymbol{\mu}$. In the next section, $\boldsymbol{\eta}$ is used for Bayesian fusion. When $\boldsymbol{\mu}$ is needed, we first compute the sparse Cholesky factorisation $Q = LL^\top$ and then solve $Q\mathbf{x} = Q(X^*, X)\mathbf{y}$ using the forward and backward substitutions. Solving such a linear system is more efficient than calculating using (5.4). In the GMRF model, the typical cost of factorising the $M \times M$ matrix $Q$ is $\mathcal{O}(M^{3/2})$ for 2D data and $\mathcal{O}(M^3)$ for 3D data.

### 5.2.2 Information-form Bayesian Fusion with Correlations

After the spatially correlated prior map $p(\xi) \sim \mathcal{N}_c(\eta, Q)$ is built, it can be updated with the observations $\mathbf{z}$ in $\Psi_2$ via the correlated Bayesian fusion. We define the likelihood function $p(\mathbf{z}|\xi)$ in the same way as in Section 3.5.3, and define its information matrix $Q_z$ to be a diagonal matrix with the entries being the inverse of constant or non-constant noise variance $\sigma^2_{z,i}$. Given the prior map and the likelihood function, we can use the MAP estimators in (3.75) and (3.76) to compute the posterior map with the information matrix $Q^+$ and information vector $\eta^+$ as follows:

$$p(\xi|\mathbf{z}) = \mathcal{N}_c(\eta^+, Q^+), \quad \text{where} \tag{5.5}$$

$$\eta^+ = \eta + H^\top Q_z \mathbf{z}, \tag{5.6}$$

$$Q^+ = Q + H^\top Q_z H. \tag{5.7}$$

Here $H$ is the observation matrix, which selects the part of state $\xi$ that is observed by $\mathbf{z}$. $H$ equals to $I_{n\times n}$ when the full state $\xi$ is observed through $\mathbf{z}$. The spatial correlations modelled in the information matrix $Q$ are used to update the neighbouring points of a single measurement.

A significant computation gain can be obtained since Bayesian fusion in the information form has a linear computational time, whereas it requires cubic time in the covariance form due to inverting the dense covariance matrix. In fact, even the most expensive computation $H^\top Q_z H$ in (5.7) is done in linear time, due to the sparsity of $Q_z$ and $H$. In addition, the addition operation in (5.7) does not make $Q^+$ become denser than $Q$. This is because the update acts to strengthen the existing constraints between measurements.

Note that there is no approximation in the fusion process described in this section since Bayesian fusion using canonical parameters and moment parameters are equivalent. The only approximation in GMRF-BF is the amount of neighbours that is considered, which is similar to the approximation done in subGPBF that considers some submaps should be conditionally independent.

### 5.2.3 Map Recovery

After the information-form posterior map is obtained, some proper techniques need to be chosen to recover the mean and variances efficiently.

The sparse linear system $Q^+\mathbf{x} = \boldsymbol{\eta}^+$ is solved to recover the mean vector $\boldsymbol{\mu}^+$. The factorisation algorithms for solving linear systems are motivated by the computational inefficiency of matrix inversion (Horn and Johnson 2012). The sparsity of $Q^+$ can reduce the computational cost of solving the linear system. When Cholesky factorisation $Q^+ = LL^T$ is performed, only the non-zero entries in the triangle Cholesky factor $L$ are computed. Then $\boldsymbol{\mu}^+$ can be obtained by solving two triangular systems. The forward substitution is applied to solve $L\mathbf{b} = \boldsymbol{\eta}^+$, and the back substitution is applied to solve $L^\top\mathbf{x} = \mathbf{b}$. In particular, Cholesky factorisation only costs $M(p^2 + 3p)$ when the $M \times M$ matrix $Q^+$ is a band matrix with bandwidth $p$, with $p \ll M$, and the forward/backward substitution costs $2Mp$. For 2.5D grid mapping with regular grid cells, $p$ is decided by $\nu$, as is shown in Appendix E. For general sparse matrices, some re-ordering approaches, e.g. nested dissection (George 1973), bandwidth reduction (Cuthill and McKee 1969) and CAMD (Chen et al. 2008), can be used to reduce fill-ins in the Cholesky factor. However, for 2D data, it takes at least $O(M^{3/2})$ to do re-ordering, and the re-ordered Cholesky factor has at least $O(M\log M)$ fill-ins. Therefore for $M > 10^6$, iterative equation solvers (Saad 2003) can be applied to obtain an approximate solution. On the other hand, the variances could be recovered efficiently using the approach proposed by Golub and Plemmons (1980).

As discussed above in this section, the computational complexity for factorising the entire matrix $Q^+$ has constrained GMRF-BF from being used for large-scale data that are more than $10^6$. To overcome this limitation, subGMRF-BF is developed and will be explained in the next chapter.

## 5.3 Summary

This chapter has proposed a generic framework, GMRF-BF, to fuse multiple sources of sensor data into a spatially correlated, probabilistic 2.5D grid map in a computationally efficient way. GMRF-BF represents a Matérn GP as a GMRF through the SPDE approach. The GMRF representation is built from one source of noisy data. The spatial correlations are modelled using a sparse information matrix, which closely approximates the corresponding covariance matrix. The predicted high-resolution map, represented in the information form, is then used as the prior for Bayesian fusion and is updated with another source of data. Bayesian fusion with the sparse information matrix allows updating of the whole map with the sparse observations within linear time. The mean and uncertainty maps are then recovered in $O(M^{3/2})$, where $M$ is the size of the final map.

GMRF-BF resembles subGPBF in Chapter 4 in that it is also an approximation method to GPBF, which takes all the spatial correlations into computation. Their primary difference is that GMRF-BF considers the approximation from a new aspect. The use of sparse information matrices in inference and Bayesian fusion in GMRF-BF provides a new perspective and some valuable insights into efficient data fusion for large-scale mapping.

# Chapter 6

# Gaussian Markov Random Fields for Bayesian Fusion with Conditionally Independent 2.5D Submapping

Previously in this thesis, Chapter 4 and 5 have developed two different approximation methods to GPBF (see Section 3.6) for building large-scale 2.5D maps when considering spatial correlations. SubGPBF in Chapter 4 has explored a globally optimal submapping strategy given the submap CI assumption. GMRF-BF in Chapter 5 uses the efficient GMRF representation to create the global map. Both the two methods indeed approximate the dense covariance matrix, which encodes spatial correlations, to reduce the computational and memory cost.

Since GMRF-BF solves for the full map and considers all correlations within one sparse information matrix, computational cost can be prohibitive for a sufficiently large amount of data. To broaden GMRF-BF for any arbitrarily big amount of data, this chapter transfers the CI submapping strategy in subGPBF into the information form by exploiting the CI property of information-form submaps. This chapter proposes **subGMRF-BF**, a submapping approach to building a near-optimal global map in almost linear time. The main contributions of this chapter are the closed-form solutions to propagate information between CI submaps before and after fusion, all in the information form. Therefore, there is a substantial reduction in computational and memory complexity for scalability when compared with GPBF, subGPBF and GMRF-BF.

## 6.1 Problem Statement and Approach Overview

The research problem remains the same as that of Chapter 4 and 5, namely, to fuse multiple sources of sensor data to build correlated, large-scale 2.5D maps efficiently. We follow the same problem statement and dataset definition in Section 4.1. In this chapter, an innovative CI mapping strategy is developed to be used together with GMRF-BF, thus named subGMRF-BF; given that there is a direct mapping between the sparsity of the information matrix and the CI elements (see Section 3.4.1). The overview of the proposed subGMRF-BF approach is similar to that of subGPBF in Section 4.1; thus it will not be repeated here. The flowchart of subGMRF-BF is shown in Figure 6.1 and a simple implementation is in Algorithm 5.

Similarly to subGPBF, subGMRF-BF considers that the submap CI property holds, although, as discussed previously, this might be an approximation in certain cases. Note that the submap CI property is again the only assumption made in the forward and backward update algorithms. If the submaps are strictly conditionally independent, the optimality of the algorithm is preserved with subGMRF-BF. In the following, we will refer to subGMRF-BF as a globally optimal approach, without repeating that the optimality is under the CI assumption.

Figure 6.2 schematically plots the elements of the full map information and covariance matrices that are obtained with subGMRF-BF and subGPBF algorithms, respectively. In Figure 6.2a, the blue areas are the elements that are obtained with subGMRF-BF. The yellow areas in Figure 6.2b are the additional non-zero entries in subGPBF which may require to be calculated. Note that the off-diagonal blocks in Figure 6.2a are strictly zero because the submaps are *conditionally independent*, while the off-diagonal blocks in Figure 6.2b are not zero because the submaps are not *marginally independent* (see Section 3.2.3). However, as was discussed in Chapter 4, it is not required to compute the off-diagonal entries in Figure 6.2b to obtain the optimal global map. We will later refer to this figure again in Section 6.2.3.

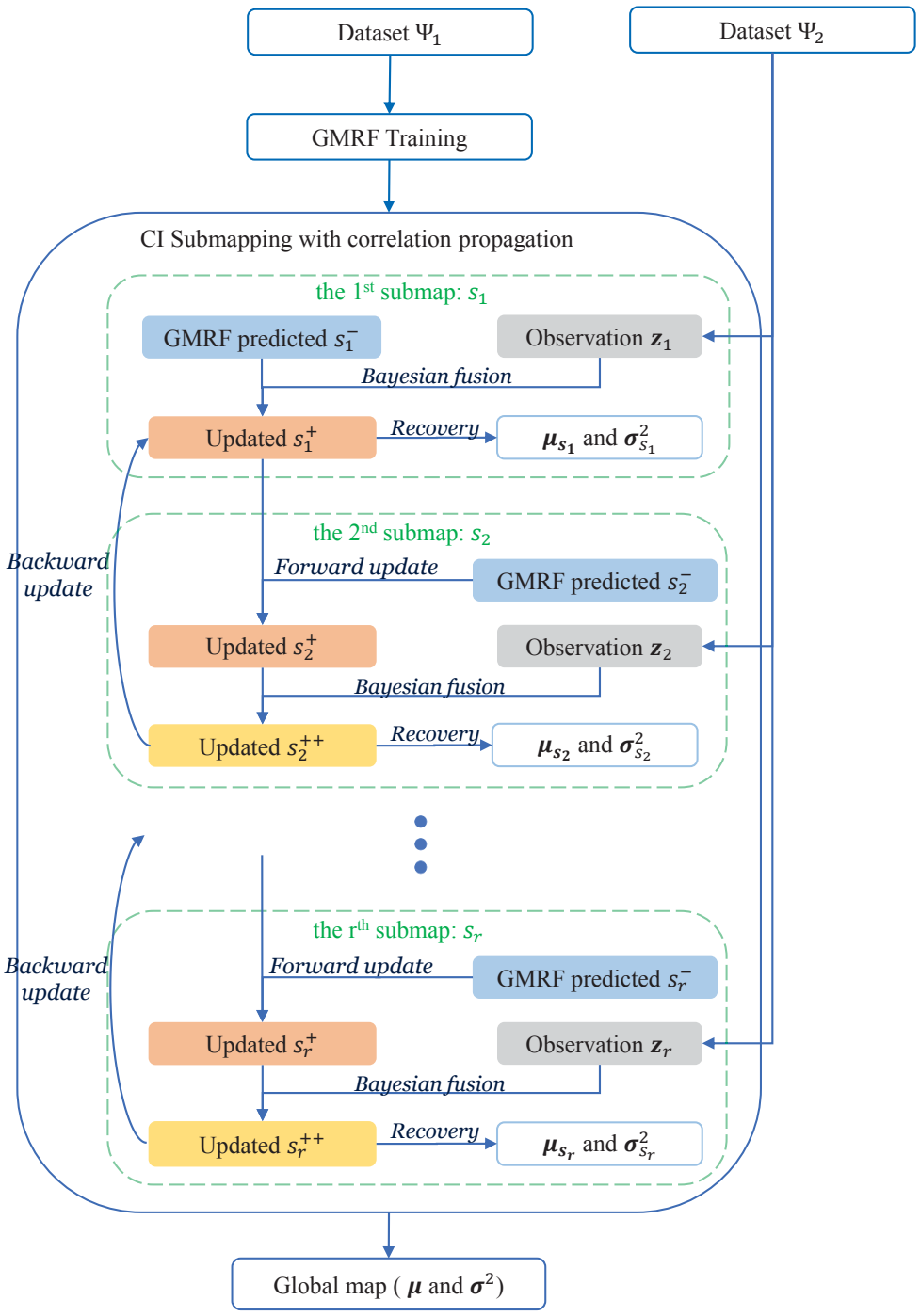



FIGURE 6.1: Flowchart of the proposed subGMRF-BF framework.

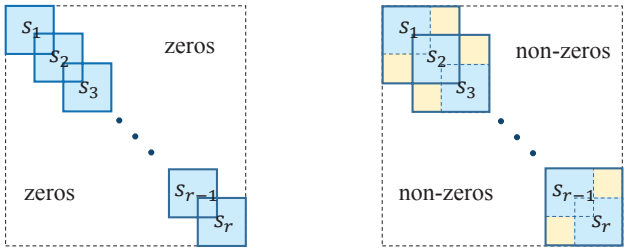



FIGURE 6.2: Schematic representation of the entries that are calculated in (A) the information matrix in subGMRF-BF, and (B) the covariance matrix in subGPBF.

**Algorithm 5** SubGMRF-BF

**Require:**

Noisy dataset 1: $\Psi_1 = \{X, \mathbf{y}\}$

Noisy dataset 2: $\Psi_2 = \{X^*, \mathbf{z}\}$ and the partitioned subsets $\{X^*_{s_i}, \mathbf{z}_{s_i}\}_{i=1}^{r}$

**Ensure:** Optimal global map

**procedure** 1 GMRF TRAINING
    Training a GMRF using $\Psi_1$
**end procedure**

**procedure** 2 FORWARD PROCESS:PRIOR MAPPING, CORRELATED FUSION AND FORWARD UPDATE
    $s_1^- = GMRFprediction(X^*_{s_1})$
    $s_1^+ = i\text{-}correlatedFusion(s_1^-, \mathbf{z}_{s_1})$ ▷ Information-form correlated Bayesian fusion
    **for** $i = 2$ to $r$ **do**
        $s_i^- = GMRFprediction(X^*_{s_i})$
        $s_i^+ = i\text{-}forwardUpdate(s_{i-1}^+, s_i^-)$ ▷ Information-form forward update (see Algorithm 6)
        $s_i^{++} = i\text{-}correlatedFusion(s_i^+, \mathbf{z}_{s_i})$ ▷ $s_i^{++}$ is currently optimal
    **end for**
    **return** $\{s_1^+, s_i^{++}\}$ where $i = 2, \ldots, r$
**end procedure**

**procedure** 3 BACKWARD UPDATE
    $s_r^{+++} \triangleq s_r^{++}$ ▷ Because $s_r^{++}$ is globally optimal given CI condition is held
    **for** $i = r$ to 2 **do** ▷ For notation simplcity, $s_1^{++} \triangleq s_1^+$
        $s_{i-1}^{+++} = i\text{-}backwardUpdate(s_i^{+++}, s_{i-1}^{++})$ ▷ Information-form backward Update (see Algorithm 7)
        $\boldsymbol{\mu}, \sigma^2 = meanRecovery(s_{i-1}^{+++})$ ▷ Recover mean and variance
    **end for**
    **return** $\{s_i^{+++}\}$ where $i = 1, \ldots, r$ ▷ Globally optimal given CI condition is held
**end procedure**

## 6.2 SubGMRF-BF: the Incremental CI Submapping Approach

### 6.2.1 The Graphical Model and Sparsity of Information Matrix

Let us consider the same graphical model as it is in Figure 4.4a, and let each node represent a set of Gaussian variables. It has been explained in Section 3.2.4 that the information matrix encodes the conditional independence of variables, while the covariance matrix models the absolute independence. In Figure 4.4a, for $i \neq j$, submap $s_i$ is correlated with submap $s_j$; accordingly, the covariance matrix is dense. By contrast, for $i \neq j$, $s_i$ and $s_j$ are *conditionally independent* given the rest of submaps[1]; hence, $Q_{s_i,s_j} = 0$ in the information matrix $Q$. Consequently, $Q$ has most of the elements being zero. The only non-zero elements are $j = i-1, i, i+1$, namely $Q$ is a block tridiagonal matrix.

Without loss of generality, the reminder of this section uses two consecutive submaps $s_1$ and $s_2$ in the same way as in Section 4.2 to illustrate the proposed CI submapping method.

### 6.2.2 GMRFs for Prior Mapping and Correlated Bayesian Fusion

SubGMRF-BF first trains a GMRF model using the SPDE approach from dataset $\Psi_1$, and then uses this model to make inference at the test points subset $X^*_{s_1}$ to create the prior submap $s_1^-$, where the superscript denotes that the estimate is produced by the GMRF model. For details about applying GMRF-SPDEs for building prior maps, please refer to Section 5.2.1. Denote the pdf of $s_1^-$ as

$$s_1^- \sim p(\xi_{s_1}). \tag{6.1}$$

Then, $s_1^-$ can be updated with the new observation $\mathbf{z}_a$ from $\Psi_2$ via the information-form Bayesian fusion, as was described in Section 5.2.2. Therefore we can get the update-to-date submap $s_1^+$, where the superscript denotes that the prior submap has been updated. Define the pdf of $s_1^+$ as

$$s_1^+ \sim p(\xi_a, \xi_b|\mathbf{z}_a) = \mathcal{N}_c(\beta^a_{s_1}, \Lambda^a_{s_1}) = \mathcal{N}_c\left(\begin{bmatrix}\beta^a_a \\ \beta^a_b\end{bmatrix}, \begin{bmatrix}\Lambda^a_a & \Lambda^a_{ab} \\ \Lambda^a_{ba} & \Lambda^a_b\end{bmatrix}\right), \tag{6.2}$$

[1]This is precisely the same independence as the pairwise Markov independence (Koller and Friedman 2009).

where the superscript represents the set of new information from $\Psi_2$ that has been incorporated into the fusion, and the subscript denotes the set of variables to be estimated.

As was explained in Section 4.2.3, the non-common observations from $\Psi_2$ are fused into each submap, except for the last submap where both the common and non-common observations are incorporated.

Note that from the second to the last submap, the observations from $\Psi_2$ will be fused with the output of the forward update, not the output of the GMRF prediction. These details are also shown in Figure 6.1 and Algorithm 5.

### 6.2.3 Information-form Forward Update

The pre-trained GMRF model is used to predict the prior submap $s_2^-$, which is defined as

$$s_2^- \sim p(\xi_b, \xi_c) = \mathcal{N}_c\left(\begin{bmatrix}\zeta_b \\ \zeta_c\end{bmatrix}, \begin{bmatrix}\Delta_b & \Delta_{bc} \\ \Delta_{cb} & \Delta_b\end{bmatrix}\right). \tag{6.3}$$

Then, the goal is to create the currently optimal submap 2, denoted as $s_2^+$, which contains the current observation $\mathbf{z}_a$. Define $s_2^+$ as

$$s_2^+ \sim p(\xi_b, \xi_c|\mathbf{z}_a) = \mathcal{N}_c\left(\begin{bmatrix}\zeta_b^a \\ \zeta_c^a\end{bmatrix}, \begin{bmatrix}\Delta_b^a & \Delta_{bc}^a \\ \Delta_{cb}^a & \Delta_c^a\end{bmatrix}\right). \tag{6.4}$$

Define the currently optimal local map[2] to be

$$p(\xi_a, \xi_b, \xi_c|z_a) = \mathcal{N}_c\left(\begin{bmatrix}\eta_a^a \\ \eta_b^a \\ \eta_c^a\end{bmatrix}, \begin{bmatrix}Q_a^a & Q_{ab}^a & Q_{ac}^a \\ Q_{ba}^a & Q_b^a & Q_{bc}^a \\ Q_{ca}^a & Q_{cb}^a & Q_b^a\end{bmatrix}\right), \tag{6.5}$$

where $Q_{ac}^a$ and $Q_{ca}^a$ are zero based on the submap CI property in (4.7). Different symbols are used in (6.2) and (6.5), although the same symbols are used in (4.3) and (4.6). This is due to the different marginalisation properties of the information-form and covariance-form MVNs, as was

[2] As was explained in Section 4.2.4, a *local map* refers to a map than contains two consecutive submaps. In this example, the local map is equal to the global map since only two submaps are considered.

explained in Section 3.2.2. In fact, the information vector and matrix in (6.2) are usually different to the corresponding blocks in (6.5).

Proposition 4.1 also applies here since it is only based on the chain rule and the submap CI property, regardless of the form of the distribution. Based on Proposition 4.1, all the information that is needed to recover the local map in (6.5) is contained in the currently optimal submap $s_1^+$ (see (6.2)) and the prior submap $s_2^-$ (see (6.3)). We propose the following forward update algorithm based on the submap CI property in (4.7). .

**Corollary 6.1** (Information-form forward update)**.** *After submap $s_1^+$ and $s_2^-$ are built, the currently optimal submap $s_2^+$ can be obtained using Part* 1 *of Algorithm 6. The information vector and matrix of $s_2^+$ are obtained as*

$$\begin{bmatrix} \zeta_b^a \\ \zeta_c^a \end{bmatrix} = \begin{bmatrix} \beta_b^a + \Delta_{bc}\Delta_c^{-1}\eta_c - \Lambda_{ba}^a(\Lambda_a^a)^{-1}\beta_a^a \\ \zeta_c \end{bmatrix}, \tag{6.6}$$

$$\begin{bmatrix} \Delta_b^a & \Delta_{bc}^a \\ \Delta_{cb}^a & \Delta_c^a \end{bmatrix} = \begin{bmatrix} \Lambda_b^a + \Delta_{bc}\Delta_c^{-1}\Delta_{cb} - \Lambda_{ba}^a(\Lambda_a^a)^{-1}\Lambda_{ab}^a & \Delta_{bc} \\ \Delta_{cb} & \Delta_b \end{bmatrix}. \tag{6.7}$$

*The local map in* (6.5) *can also be recovered using Part* 2 *of Algorithm 6 if needed.*

**Algorithm 6** Information-form forward update

**Part** 1 Getting the currently optimal submap $s_2^+$
$\zeta_b^a = \beta_b^a + \Delta_{bc}\Delta_c^{-1}\eta_c - \Lambda_{ba}^a(\Lambda_a^a)^{-1}\beta_a^a$;
$\Delta_b^a = \Lambda_b^a + \Delta_{bc}\Delta_c^{-1}\Delta_{cb} - \Lambda_{ba}^a(\Lambda_a^a)^{-1}\Lambda_{ab}^a$;
(no need for computation) $\zeta_c = \eta_c$, $\Delta_{bc}^a = \Delta_{bc}$, $\Delta_c^a = \Delta_c$.
**Part** 2 (optionally) Recovering the currently optimal local map
$\eta_b^a = \beta_b^a + \Delta_{bc}\Delta_b^{-1}\zeta_c$;
$\Delta_b^a = \Lambda_b^a + \Delta_{bc}\Delta_b^{-1}\Delta_{cb}$;
(no need for computation) $\eta_a^a = \beta_a^a$, $\eta_c^a = \zeta_c$, $\Delta_a^a = \Lambda_a^a$, $\Delta_{ab}^a = \Lambda_{ab}^a$, $\Delta_{bc}^a = \Delta_{bc}$, $\Delta_b^a = \Delta_b$.

*Proof.* See Appendix F.

Note that it is not required to recover the local optimal maps during the CI submapping process. However, should an application require the local optimal maps at any point, Part 2 of Algorithm 6 can be used.

Algorithm 2 and 6 are equivalent, as they both update the latter submap based on the difference in estimates of the shared components between both submaps. Their difference lies on the conditioning and marginalisation operations in information and covariance form (refer to Section 3.2.2). In fact, as Figure 6.2 shows, Algorithm 6 only updates the common elements between both submaps, while Algorithm 2 updates all the non-common components. In addition, when the size of submaps is fixed, it is more efficient to keep the sparse information matrices in Figure 6.2a than the dense covariance matrices in Figure 6.2b. Due to the sparsity of the information matrices, matrix multiplication and factorisation can be done very efficient in Algorithm 6. Further computation gain is achieved if the size of the common elements is smaller than that of the non-common elements, which can of course be done by design.

Submap $s_2^+$ is then fused with the new measurements $[\mathbf{z}_b; \mathbf{z}_c]$ via the information-form Bayesian fusion. Since $s_2$ is the last submap in this example, both the common and non-common observations are incorporated. Thus the globally optimal submap $s_2^{++}$ is achieved. Define $s_2^{++}$ to be

$$s_2^{++} \sim p(\boldsymbol{\xi}_{s_2}|\mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c) = \mathcal{N}_c\left(\begin{bmatrix} \zeta_b^{abc} \\ \zeta_c^{abc} \end{bmatrix}, \begin{bmatrix} \Delta_b^{abc} & \Delta_{bc}^{abc} \\ \Delta_{cb}^{abc} & \Delta_c^{abc} \end{bmatrix}\right). \tag{6.8}$$

In general, after building all submaps in the forward direction, the last submap $s_r$ is globally optimal while the other submap $s_1, \ldots, s_{r-1}$ still are not (see Section 4.2.4 for explanation). Therefore, the backward update algorithm is developed, which can correct from $s_r$ to $s_1$ at once to recover the optimal global map, i.e. mean and variances.

### 6.2.4 Information-form Backward Update

Having built the two CI submaps $s_1^+ \sim p(\boldsymbol{\xi}_{s_1}|\mathbf{z}_a)$ (see (6.2)) and $s_2^{++} \sim p(\boldsymbol{\xi}_{s_2}|\mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c)$ (see (6.8)), the aim is to compute the globally optimal submap $s_1^{++}$. Define the pdf of $s_1^{++}$ to be

$$s_1^{++} \sim p(\boldsymbol{\xi}_a, \boldsymbol{\xi}_b|\mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c) = \mathcal{N}_c\left(\begin{bmatrix} \beta_a^{abc} \\ \beta_b^{abc} \end{bmatrix}, \begin{bmatrix} \Lambda_a^{abc} & \Lambda_{ab}^{abc} \\ \Lambda_{ba}^{abc} & \Lambda_b^{abc} \end{bmatrix}\right). \tag{6.9}$$

Define the globally optimal local map to be

$$p(\boldsymbol{\xi}_a, \boldsymbol{\xi}_b, \boldsymbol{\xi}_c | \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c) = \mathcal{N}_c \left( \begin{bmatrix} \boldsymbol{\eta}_a^{abc} \\ \boldsymbol{\eta}_b^{abc} \\ \boldsymbol{\eta}_c^{abc} \end{bmatrix}, \begin{bmatrix} Q_a^{abc} & Q_{ab}^{abc} & Q_{ac}^{abc} \\ Q_{ba}^{abc} & Q_b^{abc} & Q_{bc}^{abc} \\ Q_{ca}^{abc} & Q_{cb}^{abc} & Q_c^{abc} \end{bmatrix} \right), \tag{6.10}$$

where $\Delta_{ac}^{abc}$ and $\Delta_{ca}^{abc}$ are zero based on submap CI property in (4.14).

Proposition 4.3 also applies here, since it has no limit on the density form. We then propose the following backward update algorithm based on the submap CI property in (4.14).

**Corollary 6.2** (Information-form backward update)**.** *After submap $s_1^+$ and $s_2^{++}$ are built, the globally optimal submap $s_1^{++}$ can be obtained using Part* 1 *in Algorithm 7. The information vector and matrix of $s_1^{++}$ are obtained as*

$$\begin{bmatrix} \boldsymbol{\beta}_a^{abc} \\ \boldsymbol{\beta}_b^{abc} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\beta}_a^a \\ \boldsymbol{\zeta}_b^{abc} + \Lambda_{ba}^a (\Lambda_a^a)^{-1} \boldsymbol{\beta}_a^a - \Delta_{bc}^{abc} (\Delta_c^{abc})^{-1} \boldsymbol{\zeta}_c^{abc} \end{bmatrix}, \tag{6.11}$$

$$\begin{bmatrix} \Lambda_a^{abc} & \Lambda_{ab}^{abc} \\ \Lambda_{ba}^{abc} & \Lambda_b^{abc} \end{bmatrix} = \begin{bmatrix} \Lambda_a^a & \Lambda_{ab}^a \\ \Lambda_{ba}^a & \Delta_b^{abc} + \Lambda_{ba}^a (\Lambda_a^a)^{-1} \Lambda_{ab}^a - \Delta_{bc}^{abc} (\Delta_c^{abc})^{-1} \Delta_{cb}^{abc} \end{bmatrix}. \tag{6.12}$$

*The local map in* (6.10) *can also be recovered using Part* 2 *of Algorithm 7.*

**Algorithm 7** Information-form backward update

1: **Part** 1 Getting the globally optimal submap $s_1^{++}$
2: $\boldsymbol{\beta}_b^{abc} = \boldsymbol{\zeta}_b^{abc} + \Lambda_{ba}^a (\Lambda_a^a)^{-1} \boldsymbol{\beta}_a^a - \Delta_{bc}^{abc} (\Delta_c^{abc})^{-1} \boldsymbol{\zeta}_c^{abc}$;
3: $\Lambda_b^{abc} = \Delta_b^{abc} + \Lambda_{ba}^a (\Lambda_a^a)^{-1} \Lambda_{ab}^a - \Delta_{bc}^{abc} (\Delta_c^{abc})^{-1} \Delta_{cb}^{abc}$;
4: $\boldsymbol{\beta}_a^{abc} = \boldsymbol{\beta}_a^a$, $\Lambda_a^{abc} = \Lambda_{ab}^a$, $\Lambda_{ab}^{abc} = \Lambda_{ab}^a$.
5: **Part** 2 (optionally) Recovering the globally optimal local map
6: $\boldsymbol{\eta}_b^{abc} = \boldsymbol{\zeta}_b^{abc} + \Lambda_{ba}^a (\Lambda_A^a)^{-1} \boldsymbol{\beta}_a^a$;
7: $Q_b^{abc} = \Delta_b^{abc} + \Lambda_{ba}^a (\Lambda_A^a)^{-1} \Lambda_{ab}^a$;
8: $\boldsymbol{\eta}_a^{abc} = \boldsymbol{\beta}_a^a$, $\boldsymbol{\eta}_c^{abc} = \boldsymbol{\zeta}_c^{abc}$, $Q_a^{abc} = \Lambda_a^a$, $Q_{ab}^{abc} = \Lambda_{ab}^a$, $Q_{bc}^{abc} = \Delta_{bc}^{abc}$, $Q_c^{abc} = \Delta_c^{abc}$.

*Proof.* See Appendix G.

Although Algorithm 7 can generate both the submaps and local map, in practice, we correct submaps backwards one after the other, as explained in Section 6.2.3. Algorithm 3 is more efficient than its dual in Algorithm 7. The reasons are the same as those in Section 6.2.3.

The optimality of subGMRF-BF is ensured by using the forward and backward update algorithms together. By now, the globally optimal submaps $s_1^{++}$ and $s_1^{++}$ have been obtained. Then the optimal global mean and variances can be recovered from these two submaps.

### 6.2.5 Map Recovery

After correcting all submaps, one can recover the mean and variances. Thanks to the submap CI property, the mean recovery takes place locally without solving for the total state vector of the global map. This allows opportunities for parallel and distributed computation. For each submap, its mean and variance are recovered using the method proposed in Golub and Plemmons (1980), which requires Cholesky factorisation. The re-ordering methods and the linear system solver described in Section 5.2.3 can be applied to each submap. The Cholesky factorisation typically consumes the most memory and time. Nonetheless, the small and re-ordered information matrices can be factorised in $O(m^{3/2})$, where $m$ is the size of submaps. For the bandwidth Cholesky factor with a few non-zero entries far-off the diagonal, recovery is done in linear time using the method presented by Golub and Van Loan (1996).

## 6.3 Summary

SubGMRF-BF is an efficient information-form CI submapping method, which can fuse a vast amount of sensor data from different sources in a spatially correlated and statistically sound way. In subGMRF-BF, a GMRF is first fitted to one source of sensor data; then CI submaps are inferred using this model and updated individually with the new information. The new information from either new sources of previous submaps can be propagated forward to all submaps without loss of information by using the forward update algorithm. Finally, the backward update algorithm propagates the information from the last to first submap to recover the fully updated map.

GMRF, when combined with submap CI property, significantly accelerates the submapping albeit it closely approximates the optimal global solution. The primary contribution is the derivation of the correlation propagation algorithms to optimally transmit information through CI submaps by only correcting the shared parts between consecutive submaps, which is more efficient than its dual in subGPBF. Therefore subGMRF-BF is appealing for fast, large-scale 2.5D mapping.

# Chapter 7

# Experimental Results

In this chapter, the three methods proposed in this thesis, together with the existing benchmark methods, are evaluated on two different datasets. The aim is to compare the capabilities of various methods to deal with incomplete, noisy and sparse data to generate complete and dense 2.5D maps with lower uncertainty. In addition, the computational efficiency is analysed. Both quantitative and qualitative evaluations are presented in this chapter.

## 7.1 Experimental Procedure

### 7.1.1 Comparison Approaches

**GPBF** (globally optimal): discussed in Section 3.6.

**subGPBF** (the proposed): discussed in Chapter 4.

**GMRF-BF** (the proposed): discussed in Chapter 5.

**subGMRF-BF** (the proposed): discussed in Chapter 6.

**GPBF-BCM**: GPBF with BCM (Schwaighofer and Tresp 2003) for submapping. This is an independent submapping approach since the spatial correlations between submaps are ignored during prediction, yet the correlations within each submap are considered during both inference and fusion. We implemented the standard BCM with local GP experts (see Section 2.4.2) using

the software provided by the author of BCM (Schwaighofer 2005). Local GPs are trained with independent subsets from $\Psi_1$, then each submap is predicted by weighting the average on all local GP predictions. Such BCM model is applied to create the prior submaps, which are then updated with the non-common observations from $\Psi_2$ using the correlated Bayesian fusion. GPBF-BCM includes correlations within fusion while Jadidi et al. (2014) did naïve fusion to reduce computational cost. For building prior submaps, GPBF-BCM combines all local GP predictions, while Kim and Kim (2013) considered only the local GPs whose training subsets overlapped; yet the latter set overlapping submaps to avoid discontinuous boundaries. None of these three approaches allows knowledge propagation between submaps.

**CCIS**: subGPBF without the forward update, thereby the global optimality cannot be guaranteed. During submapping, it is always the submap inferred by GPs that is updated in Bayesian fusion. CCIS was proposed in our previous work (Sun et al. 2015).

**ICIS**: subGMRF-BF without the forward update, thereby the global optimality cannot be guaranteed. During submapping, it is always the submap inferred by GMRF-SPDEs that is updated in Bayesian fusion. ICIS was proposed in our previous work (Sun et al. 2017).

**sparseGPBF**: GPBF with a compactly supported piecewise polynomial kernel $K_{pp,2}(r)$ (see (3.56)). Such kernel generates a sparse covariance matrix for the training data, thereby allowing the hyperparameters to be learned efficiently. However, GP prediction generates a dense covariance matrix after conditioning, which makes Bayesian fusion expensive. The predicted map is then updated with the observations in $\Psi_2$. One measurement can update a local area by using the correlations encoded in the covariance matrix. SparseGPBF is similar to GMRF-BF in the sense that only correlations between adjacent points are kept. However, the sparse kernel cuts the correlations in a hard way while GMRF-BF learns the best approximation. Therefore the GMRF-BF captures the natural spatial correlations better than the static sparse covariance kernel.

**naïveGPBF**: GP and the naïve Bayesian fusion (see Section 3.5.2) for global mapping. The same prior map in GPBF is used for fusion. The computational complexity of naïveGPBF is constant. However, point measurements are not propagated to the neighbouring points since there are no correlations between them. This comparison is mainly to show how the spatial correlations improve the accuracy and smoothness of the fusion result.

### 7.1.2 Datasets and Sensor Information

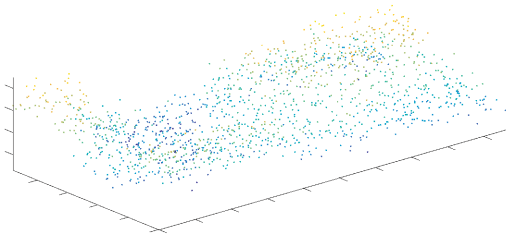
#### 7.1.2.1 Terrain Dataset (Synthetic Noisy Data)

A challenging scenario is simulated considering data incompleteness, uncertainty and inconsistency. Synthetic elevation data is generated from Canadian Digital Elevation Data (CDED) (Natural Resources Canada. Laval, Quebec 1995). CDED consists of the ground elevations at regularly spaced grids. The data synthetic process follows that of Gerardo-Castro et al. (2015).

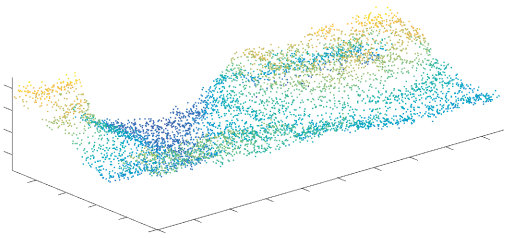
We randomly choose one map from CDED, which covers the area of $[-116°, -114°]$ longitude and $[56°, 57°]$ latitude. There are $9608 \times 4804$ grid cells in this map, and each cell is $23\text{m} \times 23\text{m}$. The elevation values range between $[430\text{m}, 850\text{m}]$. To generate synthetic data, we randomly sample 1476 points from the original map and set noise level to be $\sigma_\epsilon = 50.4\text{m}$ (12% of the elevation range) to get $\Psi_1$, which is regarded as sparse measurement. We also randomly sample 5832 points from the original map and set the noise level to be $\sigma_z = 12.6\text{m}$ (3% of the elevation range) to obtain $\Psi_2$, which contains dense data. Please notice that both $\Psi_1$ and $\Psi_2$ are **incomplete**. To get the groundtruth (GT), the original map is down-sampled into a $100 \times 200$ grid map. $\Psi_1$, $\Psi_2$ and the groundtruth are shown in Figure 7.1. Note in the 3D plots of Figure 7.1a and Figure 7.1b, we enlarge the axis ratio of the third dimension to make the elevation change visible. In the 2.5D plots of elevation data in this Chapter, the vertical axis corresponds to the latitude and horizontal axis is the longitude, and the elevation values and uncertainty are shown in colour. The colour axis range is $[430\text{m}, 850\text{m}]$ for elevation values, e.g. Figure 7.1e, and $[0\text{m}, 100\text{m}]$ for uncertainty, e.g. Figure 7.3b and 7.3d.

#### 7.1.2.2 Pipe Wall Thickness Dataset (Real Experimental Data)

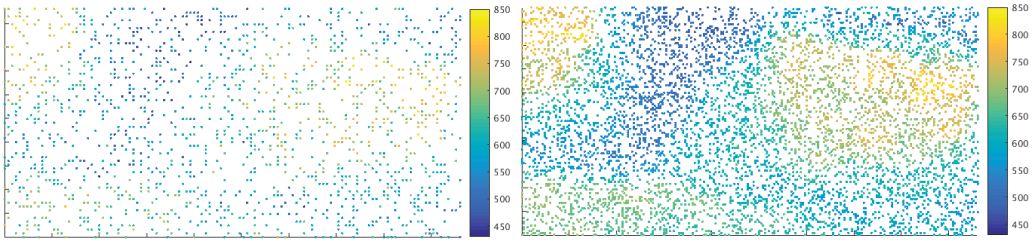
Experiments have also been done to build complete, dense and high-resolution 2.5D remaining wall thickness maps of water pipes, to aid the condition assessment of pipes. Such dataset contains the remaining wall thickness values measured on pipes' surfaces. Electromagnetic sensors can be used to get the thickness values. In this experiment, two data sources of pipe's remaining wall thickness measurements are fused: (1) a pulsed-Eddy current sensor (Ulapane et al. 2014) which collects complete measurements with higher uncertainty, and (2) a magnetic flux leakage sensor (Wijerathna et al. 2013) which gives incomplete measurements with lower uncertainty.



(A) $\Psi_1$ data points in 3D view

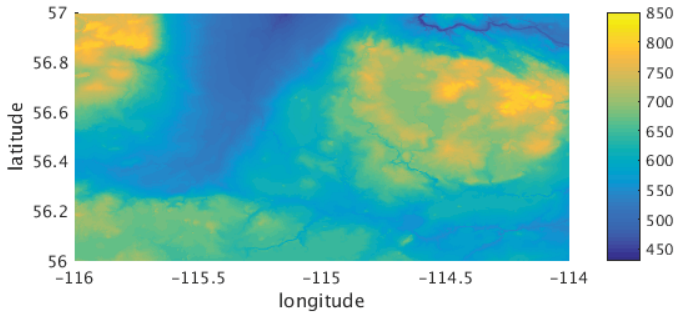


(B) $\Psi_2$ data points in 3D view




(C) $\Psi_1$ data points in 2D view

(D) $\Psi_2$ data points in 2D view



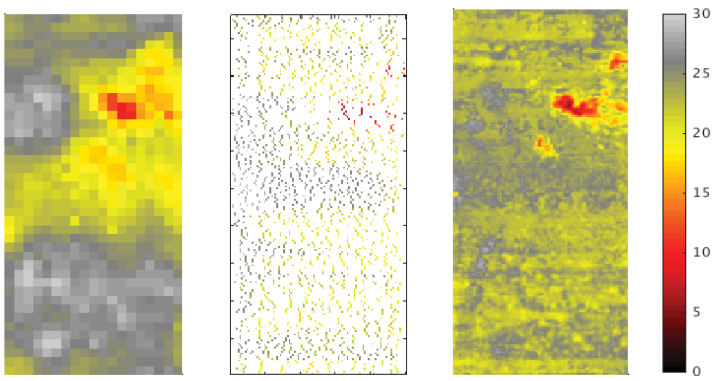

(E) Groundtruth (GT)

FIGURE 7.1: The synthetic terrain datasets and groundtruth.

We use pulsed-Eddy current sensor data as $\Psi_1$ and magnetic flux leakage sensor data as $\Psi_2$. Both sensor data have iid noises. Please be aware that $\Psi_2$ has some missing data, and only the available data is used during fusion. The GT data comes from ray tracing high-accuracy laser data from the internal and external surfaces (Skinner et al. 2014).

All sensor measurements were taken from a real cast-iron pipe section of 1000 mm length and 2073.5 mm circumference. The pulsed-Eddy current sensor measured the thickness values at all the 42×20 grids, as is shown by Figure 7.2a. The magnetic flux leakage sensor only collected 1614

meaningful measurements from the $192 \times 100$ grids, as Figure 7.2b shows that the measurements are discontinuous as dots. The GT measurements covers the full circumference and the 0 to 891.8mm length, and size of each grid in the map is $1.2\text{mm} \times 1.2\text{mm}$. Therefore quantitative comparisons with GT are done from 0 to 891.8mm length and the whole circumference. The GT thickness map is shown in Figure 7.2c. In the 2.5D plots, thickness and uncertainty are shown in colour. The colour axis range of the thickness map is from 0 to $30mm$, and that of the uncertainty is from $0mm$ to $8mm$. The Cylindrical coordinates of the real pipe have been converted into Cartesian coordinates, i.e. the vertical axis corresponds to the circumferential axis of the pipe, and the horizontal axis is the longitudinal axis.



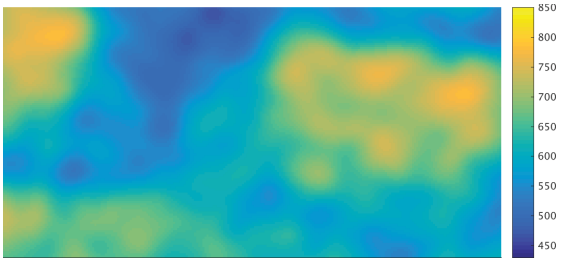

(A) data $\Psi_1$ (B) data $\Psi_2$ (C) GT

FIGURE 7.2: Pipes' remaining wall thickness maps in 2D view.

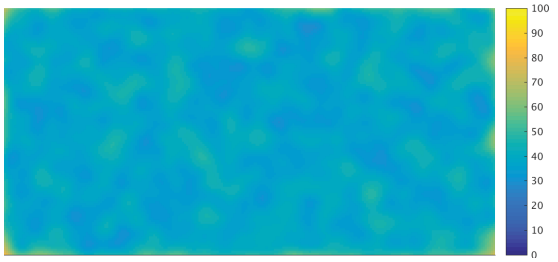
### 7.1.3 Evaluation

This subsection describes the general methodology applied to the datasets. All the compared approaches described in Section 7.1.1 have been tested on the datasets in Section 7.1.2. All tests are done on a workstation with sixteen 3.10 Ghz Intel Xeon E5-2687W processors. A 5-run Monte-Carlo simulation is performed on the terrain dataset during the evaluation.

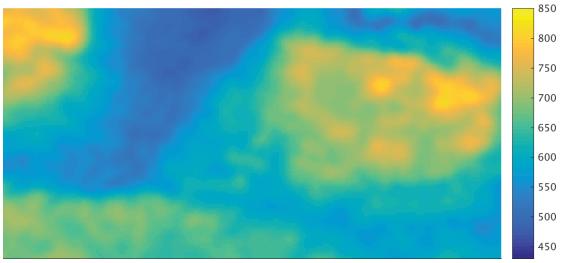
Either $\Psi_1$ or $\Psi_2$ can be used to learn the hyperparameters and build the prior map. As was mentioned in Section 3.3 and 3.4, aided by the power of GP, GPs or GMRF-SPDEs can increase or decrease the grid resolution of maps by inference. They will fill the gaps in the sensor data with probabilistic values that are correlated with neighbouring areas covered by the sensors. Here $\Psi_1$ is used as training data to predict the same resolution map ($192 \times 100$) as $\Psi_2$. Then, the correlated



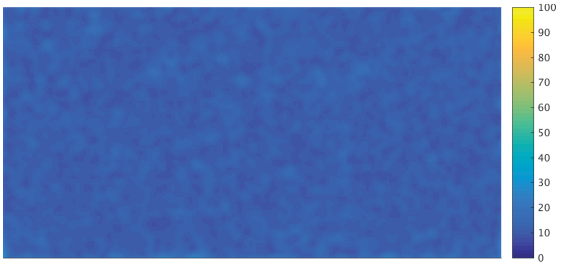

(A) Prior map: $\mu$ (predicted by GP)




(B) Prior map: $2\sigma$ (predicted by GP)

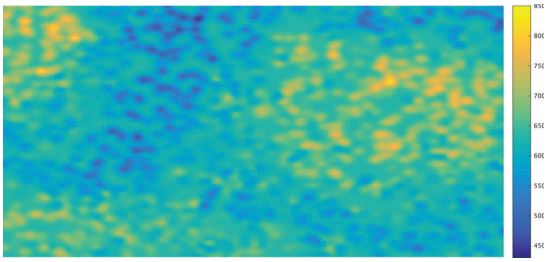



(C) Global fusion result: $\mu$

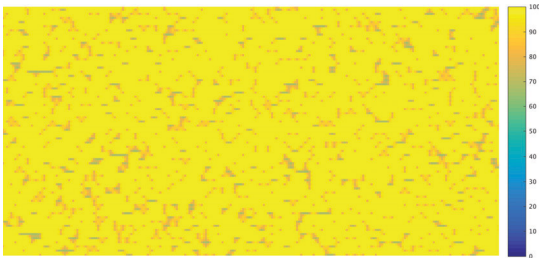



(D) Global fusion result: $2\sigma$

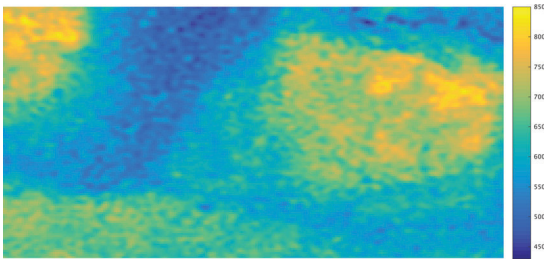
FIGURE 7.3: GPBF (elevation data)



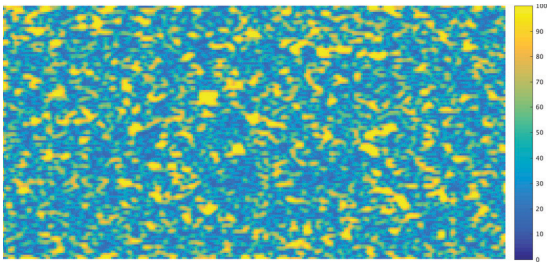
(A) Prior map: $\mu$ (predicted by sparse GP)



(B) Prior map: $2\sigma$ (predicted by sparse GP)



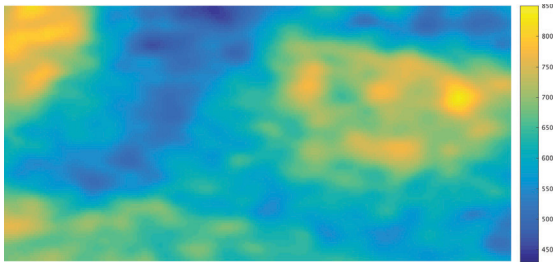
(C) Global fusion result: $\mu$



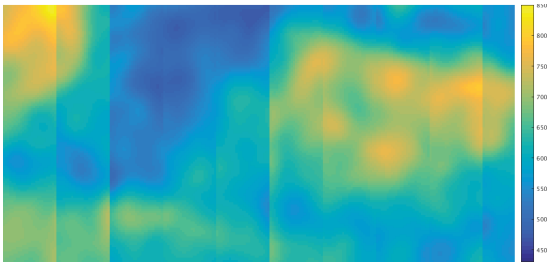
(D) Global fusion result: $2\sigma$

FIGURE 7.4: SparseGPBF (elevation data)

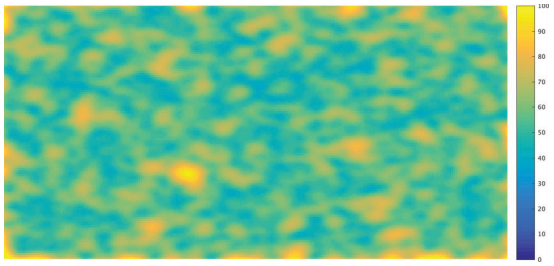
fusion is performed at the high resolution, and vice-versa whenever it is computationally feasible. Since we focus on how to handle a large amount of test data, we train the GP or GMRF-SPDE using $\Psi_1$. For training with a huge amount of data, the approximation methods described in Section 2.4 can be used. Therefore the same global GP model is used for GPBF, subGPBF and CCIS; the same GMRF-SPDE model is used for GMRF-BF, subGMRF-BF and ICIS.



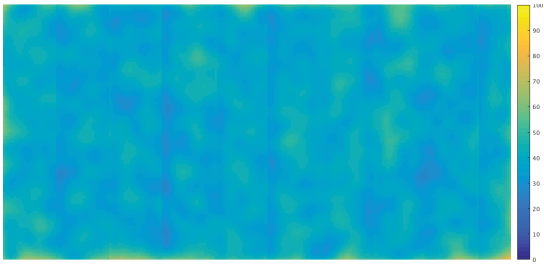
(A) Prior map: $\mu$ (predicted by GMRF-SPDE)



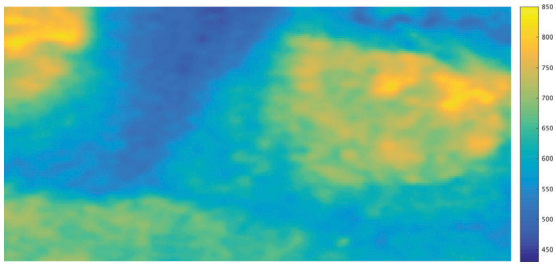
(A) Prior map: $\mu$ (predicted by GP-BCM)



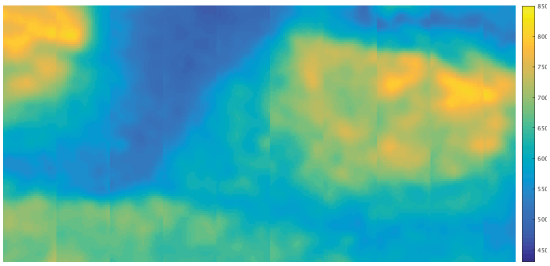
(B) Prior map: $2\sigma$ (predicted by GMRF-SPDE)



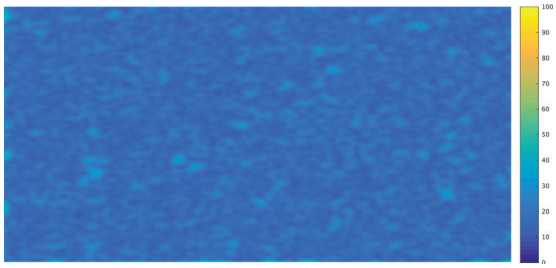
(B) Prior map: $2\sigma$ (predicted by GP-BCM)



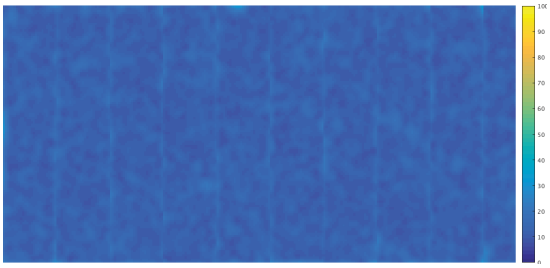
(C) Global fusion result: $\mu$



(C) Submap-level fusion result: $\mu$



(D) Global fusion result: $2\sigma$

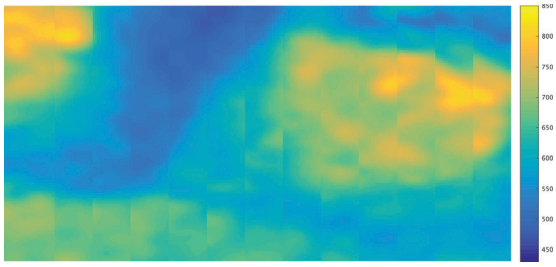


(D) Submap-level fusion result: $2\sigma$

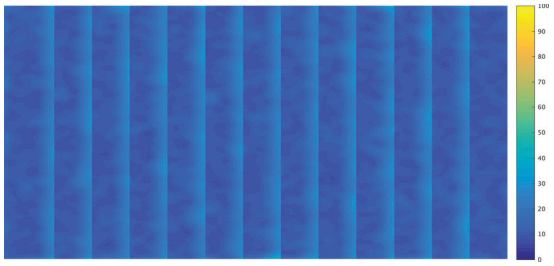
FIGURE 7.5: GMRF-BF (elevation data)

FIGURE 7.6: GPBF-BCM (elevation data)

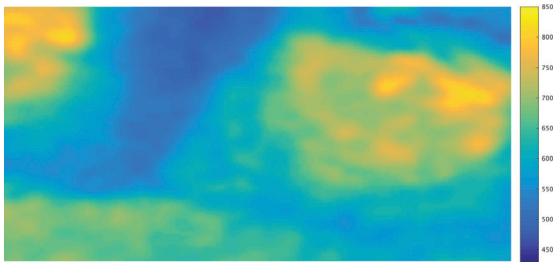
For all the compared approaches that use a GP, we set a zero mean function and a Matérn covariance function with $\nu = 3/2$. The hyperparameters are learned via optimising the log-likelihood function. GP is implemented using the GPML toolbox (Rasmussen and Nickisch 2010). For all the compared approaches that use a GMRF-SPDE, we build the GMRF model that maps the GP model previously mentioned. We also set $\nu = 3/2$, and $\nu$ directly decides the neighbourhood size



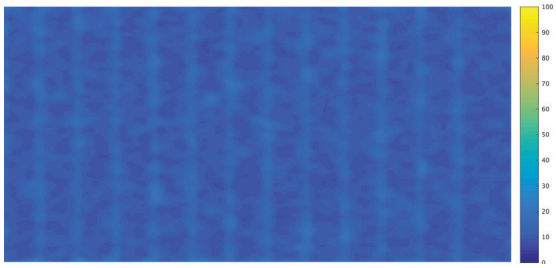
(A) Submap-level fusion result: $\mu$



(B) Submap-level fusion result: $2\sigma$

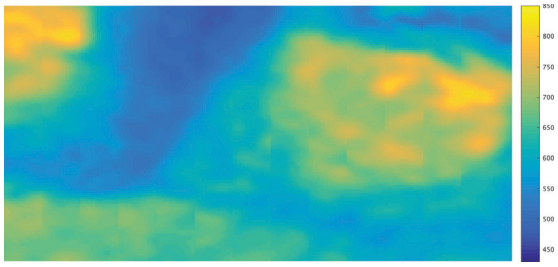


(C) Backward update result: $\mu$

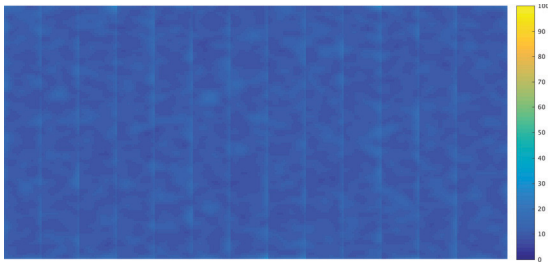


(D) Backward update result: $2\sigma$

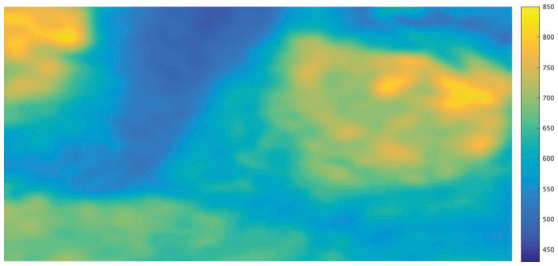
FIGURE 7.7: CCIS (elevation data)



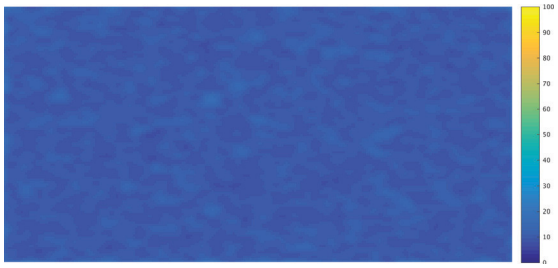
(A) Forward update result: $\mu$



(B) Forward update result: $2\sigma$



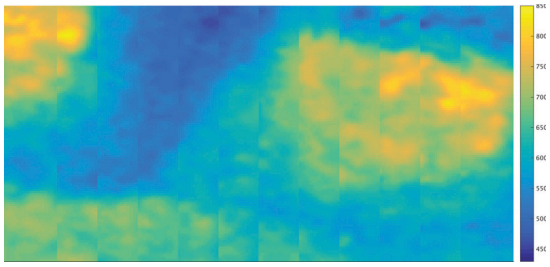
(C) Backward update result: $\mu$



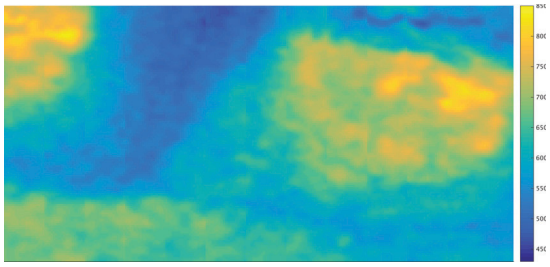
(D) Backward update result: $2\sigma$

FIGURE 7.8: SubGPBF (elevation data)

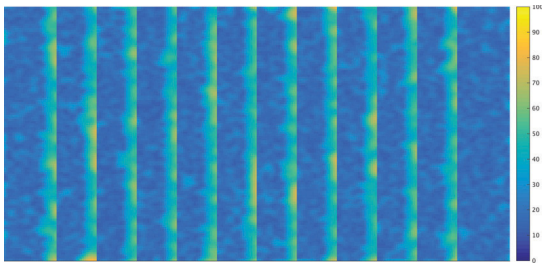
in the GMRF model (see Appendix E for details). The prior mapping using the GMRF is implemented using the R-INLA package (Lindgren and Rue 2015). As was mentioned in Section 3.4.2, R-INLA approximates $\xi$ with piecewise linear basis functions and then gets the GMRF by solving the SPDE. The predictive mean is obtained by the sum of linear predictor components. There are two ways to compute the predictive information matrix, as are explained in Appendix E. The first approach only suits data at uniform grid cells while the second approach can also handle the data



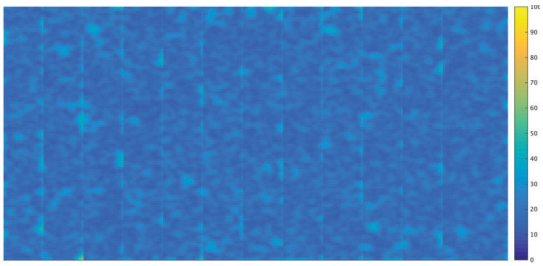
(A) Submap-level fusion result: $\mu$



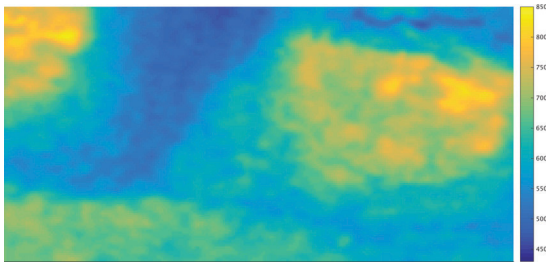
(A) Forward update result: $\mu$



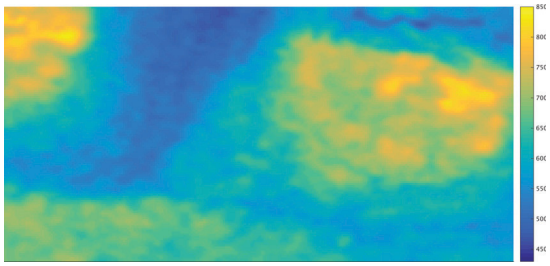
(B) Submap-level fusion result: $2\sigma$



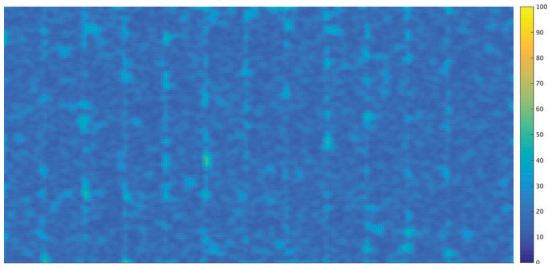
(B) Forward update result: $2\sigma$



(C) Backward update result: $\mu$

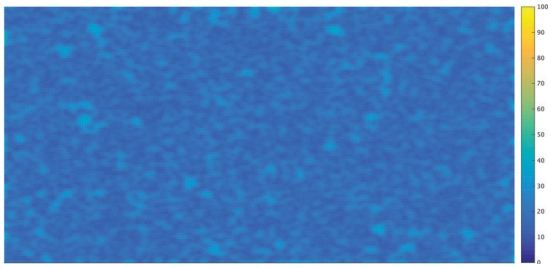


(C) Backward update result: $\mu$



(D) Backward update: $2\sigma$

FIGURE 7.9: ICIS (elevation data)

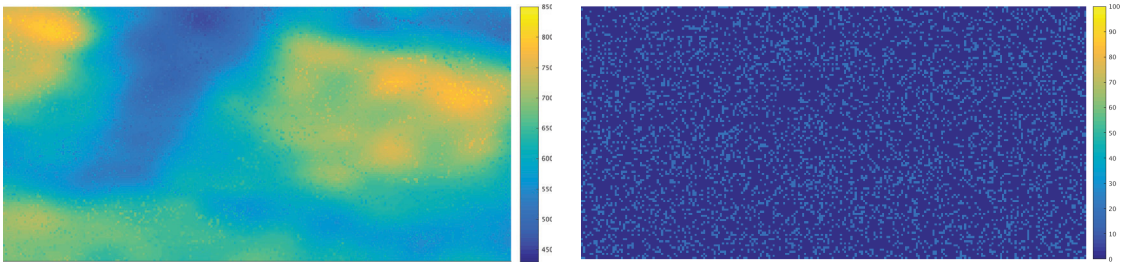


(D) Backward update result: $2\sigma$

FIGURE 7.10: SubGMRF-BF (elevation data)

at irregular locations. Here we use the second approach. This approach directly generates the information matrix of triangular basis functions, which is in the triangulation domain; therefore we need to project the information matrix to the real domain, where query points are, to get the predictive information matrix for the query points.

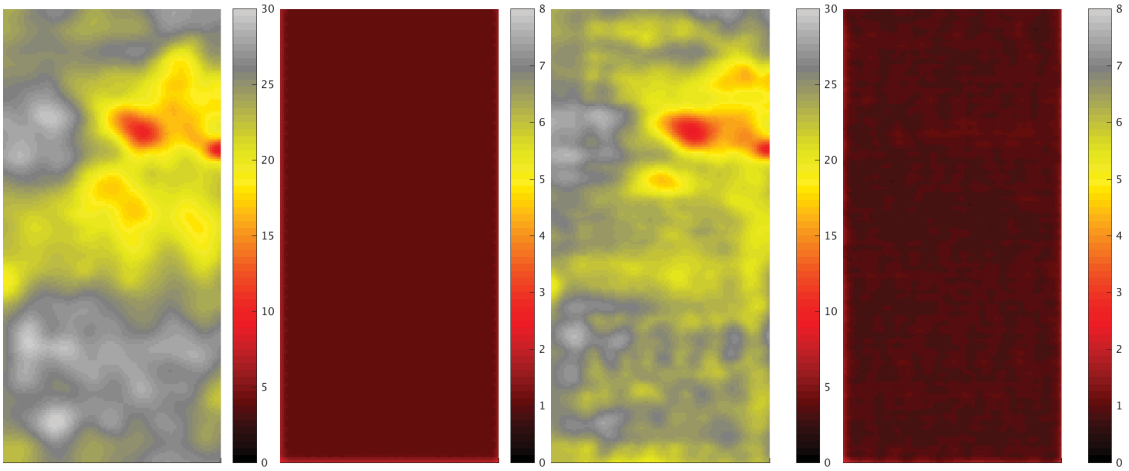
After the prior map is obtained at the desired resolution, the observations in $\Psi_2$ are used to update



(A) Naïve Bayesian fusion result: $\mu$

(B) Naïve Bayesian fusion result: $2\sigma$

FIGURE 7.11: NaïveGPBF (elevation data). The prior map is provided by GPBF.



(A) Prior map: $\mu$ (predicted by GP)

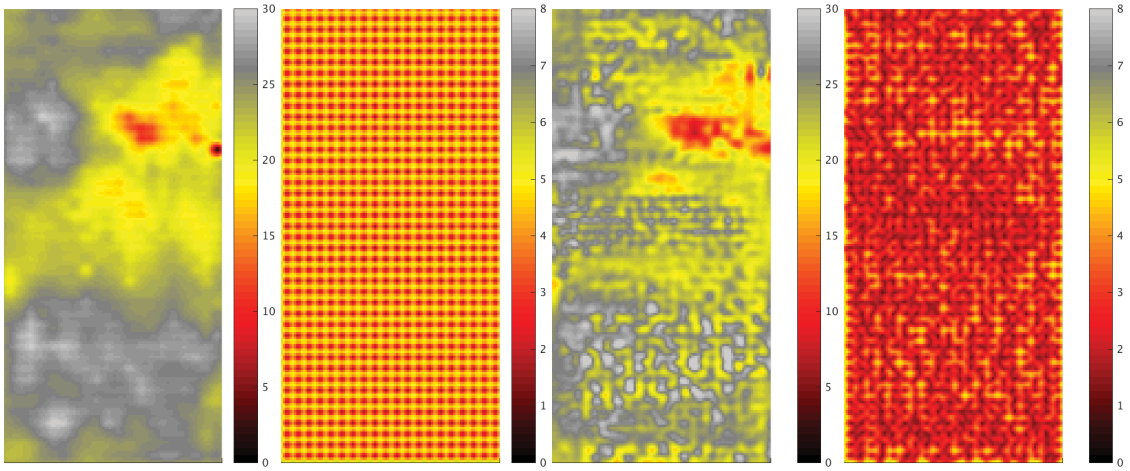
(B) Prior map: $2\sigma$ (predicted by GP)

(C) Global fusion result: $\mu$

(D) Global fusion result: $2\sigma$

FIGURE 7.12: GPBF (pipe wall thickness data)

it via Bayesian fusion. The observation matrix $H$ is created to establish the relationship between the data in $\Psi_2$ and the prior map. After fusion, different sources of data, which may be incomplete, sparse and noisy, are integrated into a complete, dense and probabilistic map with lower uncertainty. For the global mapping methods, i.e. GPBF, GMRF-BF, sparseGPBF, naïveGPBF, the final maps are obtained after the global fusion. For GPBF-BCM, which builds independent submaps, the final map is obtained after the submap-level Bayesian fusion. With the four CI submapping methods, i.e. subGPBF, subGMRF-BF, CCIS and ICIS, all of them use the backward update and the former two, subGPBF and subGMRF-BF use the forward update algorithims. As was explained in Chapter 4 and 6, subGPBF and subGMRF-BF fuse the measurements in $\Psi_2$ with the outputs of the forward update algorithms; and the final map is obtained after the backward update process. For CCIS and ICIS, measurements in $\Psi_2$ are fused with the submaps predicted by a GP or a



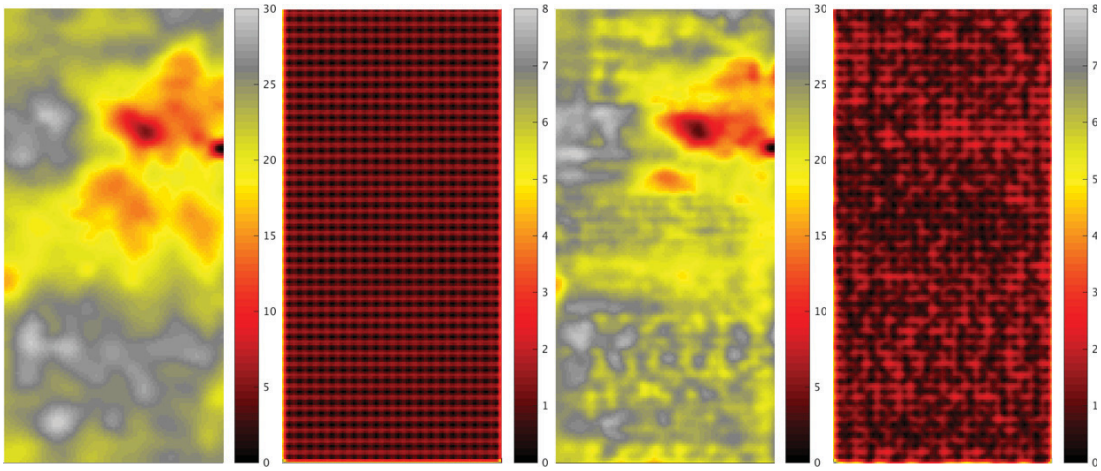
(A) Prior map: $\mu$ (predicted by sparse GP)

(B) Prior map: $2\sigma$ (predicted by sparse GP)

(C) Global fusion result: $\mu$

(D) Global fusion result: $2\sigma$

FIGURE 7.13: SparseGPBF (pipe wall thickness data)



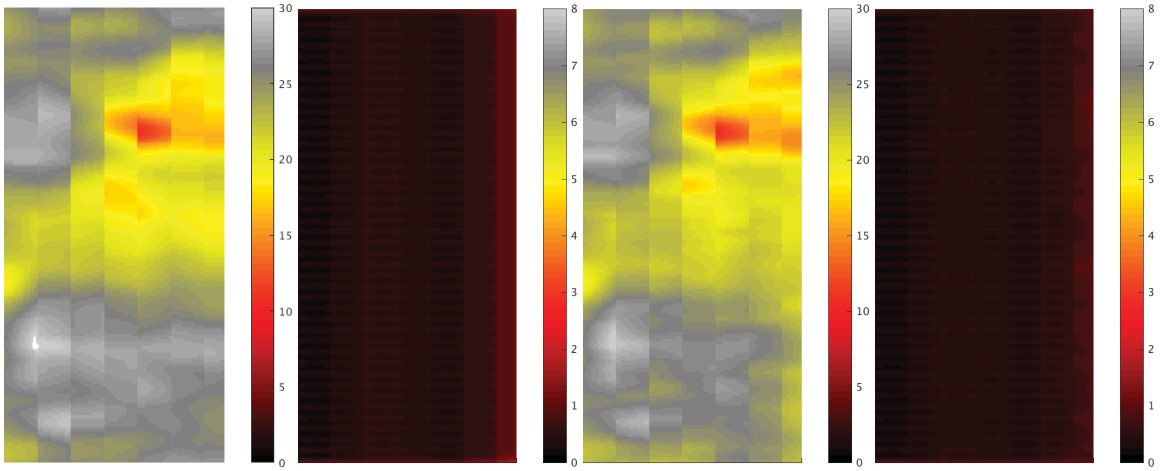
(A) Prior map: $\mu$ (predicted by GMRF-SPDE)

(B) Prior map: $2\sigma$ (predicted by GMRF-SPDE)

(C) Global fusion result: $\mu$

(D) Global fusion result: $2\sigma$

FIGURE 7.14: GMRF-BF (pipe wall thickness data)



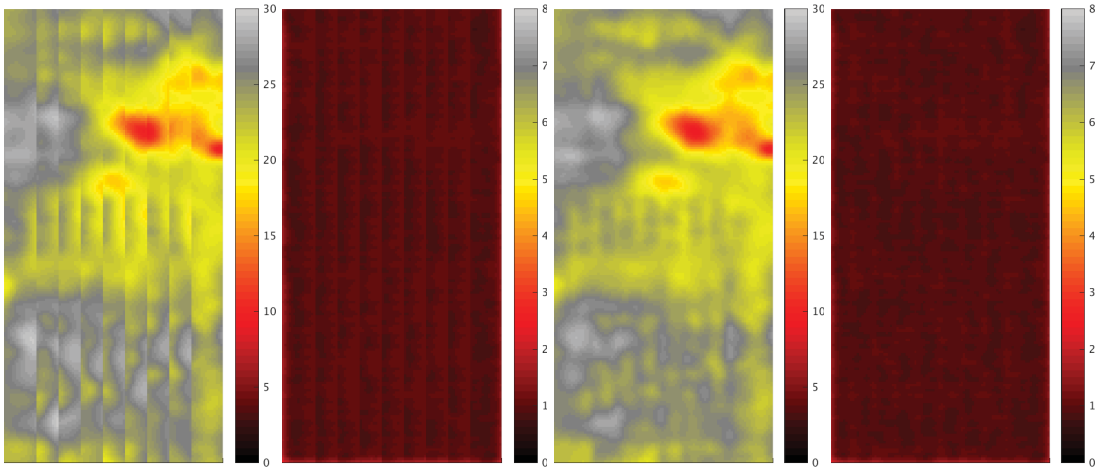
(A) Prior map: $\mu$ (predicted by GP-BCM)

(B) Prior map: $2\sigma$ (predicted by GP-BCM)

(C) Submap-level fusion result: $\mu$

(D) Submap-level fusion result: $2\sigma$

FIGURE 7.15: GPBF-BCM (pipe wall thickness data)



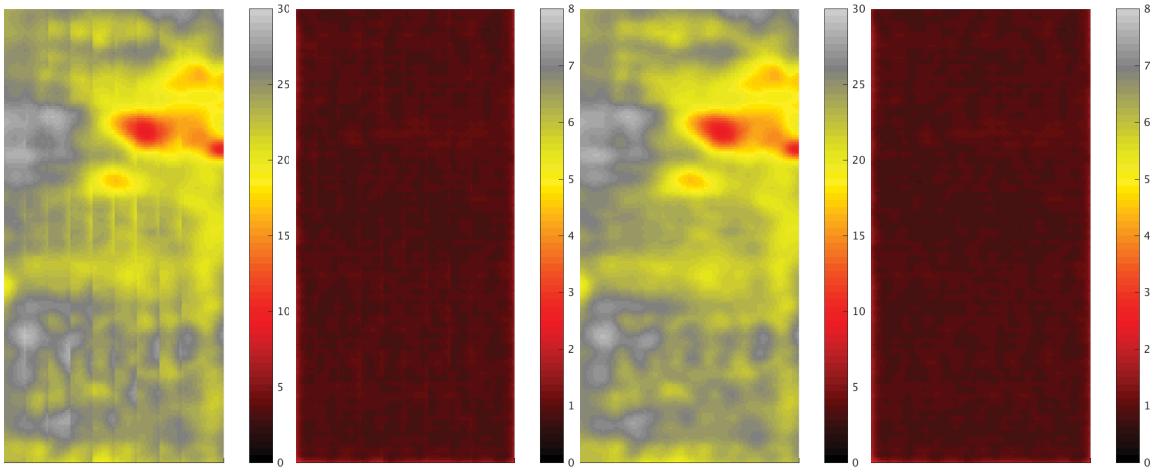
(A) Submap-level fusion result: $\mu$

(B) Submap-level fusion result: $2\sigma$

(C) Backward update result: $\mu$

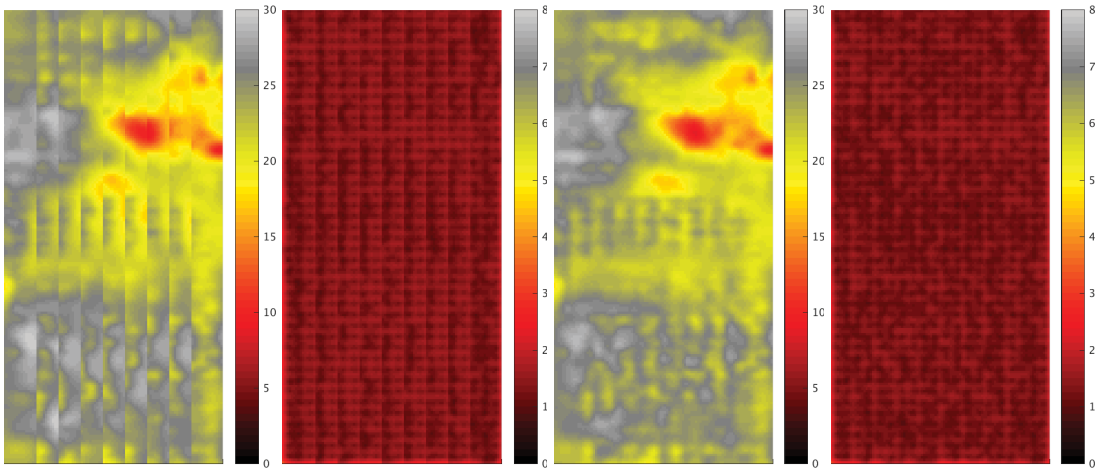
(D) Backward update result: $2\sigma$

FIGURE 7.16: CCIS (pipe wall thickness data)



(A) Forward update result: $\mu$ (B) Forward update result: $2\sigma$ (C) Backward update result: $\mu$ (D) Backward update result: $2\sigma$

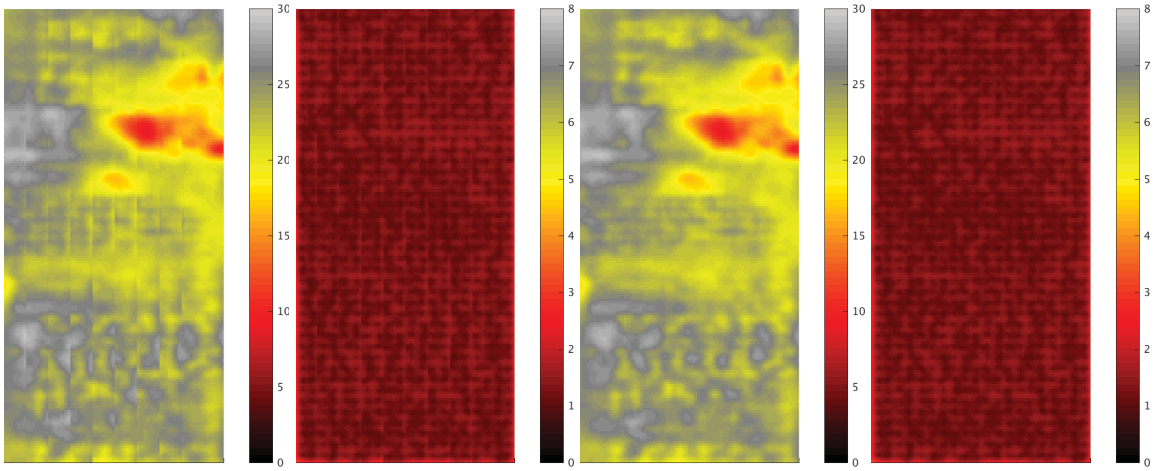
FIGURE 7.17: SubGPBF (pipe wall thickness data)



(A) Submap-level fusion result: $\mu$ (B) Submap-level fusion result: $2\sigma$ (C) Backward update result: $\mu$ (D) Backward update result: $2\sigma$

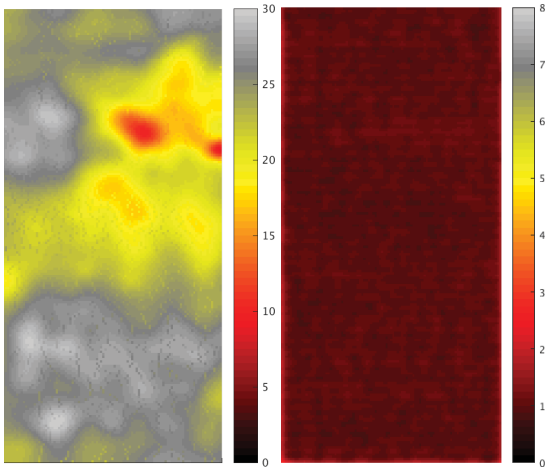
FIGURE 7.18: ICIS (pipe wall thickness data)

GMRF; and the final map is obtained after the backward update. The size of submaps is chosen to be equal and can be decided by looking at the values of the cross-correlations. The main idea is to minimise the cross-correlations that are left out by counting the number of columns until they are close to zero, therefore minimising the approximation for the proposed algorithms. Finally, in the three information-form approaches, namely GMRF-BF, subGMRF-BF and ICIS, the mean and the variances can be recovered efficiently using the methods described in Section 5.2.3



(A) Forward update result: $\mu$ (B) Forward update result: $2\sigma$ (C) Backward update result: $\mu$ (D) Backward update result: $2\sigma$

FIGURE 7.19: SubGMRF-BF (pipe wall thickness data)



(A) Naïve Bayesian fusion result: $\mu$ (B) Naïve Bayesian fusion result: $2\sigma$

FIGURE 7.20: NaïveGPBF (pipe wall thickness data). The prior map is provided by GPBF.

and Section 6.2.5 after we obtain the information-form maps. In particular, we implement multi-threaded parallel programs for submap recovery in subGMRF-BF and ICIS to greatly reduce the computational cost.

## 7.2 Results

Both qualitative and quantitative comparisons are done on the two datasets described above.

### 7.2.1 Qualitative Evaluation

Experimental results of the terrain data are plotted in Figure 7.3 - 7.11. The results on pipe wall thickness dataset are shown in Figure 7.12 - 7.20.

For the prior mapping, both GP and GMRF regression is shown to be able to learn from a small amount of incomplete data and predict complete, dense probabilistic maps, e.g. Figure 7.3a and 7.5a, Figure 7.12a and 7.14a. In addition, these prior maps correctly model the global trend, e.g. Figure 7.3a and 7.5a clearly show the bottom-right peak and mid-left valley. However, as Figure 7.3a shows, the GP predicted map sometimes tends to be smoothed. This is because the training data is of limited amount while covering a large area, and the standard GP considers far-away cross-correlations. The predicted prior maps in subGPBF and subGMRF-BF are not shown here because, based on the marginalisation property of MVNs, the value of each predicted submap is equal to the value in the globally predicted prior map in GPBF and GMRF-BF. In GPBF-BCM, the prior submaps (see Figure 7.6a and 7.15a) have discontinuous boundaries due to the hard cut-off of training and test data in BCM. In fact, although the predicted values are based on all training subsets, the correlations between the training and test subsets are lost during BCM prediction and Bayesian fusion. BCM is an optimistic approach, therefore the uncertainty is smaller than that of GPBF, as are shown in Figure 7.6b and 7.15b.

For the Bayesian fusion, Figure 7.3c, 7.5a, 7.12c and 7.14a show that the consistency has been improved by considering spatial correlations during fusion in contrast to the naïveGPBF, which generates scattered points result (see in Figure 7.11 and 7.20). SparseGPBF only includes the spatial correlations within a local area, thereby one single measurement only updates a small number of neighbours (see Figure 7.4 and 7.13). Independently of the comparison method, the uncertainty is reduced after fusion. In the submap-based methods without forward update, namely GPBF-BCM, CCIS and ICIS, the fusion results have some inconsistent boundary areas (see Figure 7.6c, 7.6d, 7.7a, 7.7b, 7.9a and 7.9b) since the non-overlapping observations from $\Psi_2$ are fused within

each submap. In contrast, the forward update algorithm enables the current submap to be always up-to-date, as in Figure 7.8a, 7.10a, 7.17a and 7.19a. There figures show that even without the backward update algorithm, the mean maps produced by the forward update algorithm are already very close to the final results. However, there are some clear boundary lines between submaps in the uncertainty maps after the forward update, see Figure 7.8b, 7.10b, 7.17b and 7.19b. This is because we plot the uncertainty submaps obtained after both the forward update and submap fusion; and it is natural that the boundary areas have higher uncertainty since they contain less correlated observations compared with the central areas.

The final step in CCIS, subGPBF, ICIS and subGMRF-BF is backward update. The results on elevation data (see Figure 7.7c, 7.8c, 7.9c, 7.10c) and thickness data (see 7.16c, 7.17c, 7.18c, 7.19c) show that the backward update algorithm has improved accuracy of the mean estimates, which closely approximate the globally optimal estimates produced by GPBF, as in Figure 7.3c and 7.12. In addition, the backward update algorithm reduces uncertainty and improves consistency in the boundaries between submaps, as shown in the elevation maps (see Figure 7.7d, 7.8d, 7.9d, 7.10d) and the thickness maps (see Figure 7.16d, 7.17d, 7.18d, 7.19d). These figures also show that subGPBF and subGMRF-BF approximate very well GPBF and GMRF-BF, respectively.

Table 7.1: Computational complexity of all compared methods

| | Training | Prediction | Fusion | Propagation | Recovery |
|---|---|---|---|---|---|
| **subGPBF** | $\mathcal{O}(N^3)$ | $\mathcal{O}(N^3+N^2m)$ | $\mathcal{O}(Mm^2)$ | $\mathcal{O}(Mm^2)+\mathcal{O}(Mm^2)$ | / |
| **GMRF-BF** | $\mathcal{O}(N^{3/2})$ | $\mathcal{O}(M)$ | $\mathcal{O}(M)$ | / | $\mathcal{O}(M^{3/2})$ |
| **subGMRF-BF** | $\mathcal{O}(N^{3/2})$ | $\mathcal{O}(m)$ | $\mathcal{O}(M)$ | $\mathcal{O}(M\sqrt{m})+\mathcal{O}(M\sqrt{m})$ | $\mathcal{O}(m^{3/2})$ |
| GPBF | $\mathcal{O}(N^3)$ | $\mathcal{O}(N^3+N^2M)$ | $\mathcal{O}(M^3)$ | / | / |
| GPBF-BCM | $\mathcal{O}(\frac{N^3m}{M^2})$ | $\mathcal{O}(\frac{N^3m}{M}+N^2m+Mm^2)$ | $\mathcal{O}(Mm^2)$ | / | / |
| CCIS | $\mathcal{O}(N^3)$ | $\mathcal{O}(N^3+N^2m)$ | $\mathcal{O}(Mm^2)$ | $\mathcal{O}(Mm^2)$ | / |
| ICIS | $\mathcal{O}(N^{3/2})$ | $\mathcal{O}(M)$ | $\mathcal{O}(M)$ | $\mathcal{O}(M\sqrt{m})$ | $\mathcal{O}(m^{3/2})$ |
| sparseGPBF | $\mathcal{O}(N^3)$ | $\mathcal{O}(N^3+N^2M)$ | $\mathcal{O}(M^3)$ | / | / |
| naïveGPBF | / | / | $\mathcal{O}(1)$ | / | / |

$N$, $M$ and $m$ are the size of training data, test data, and submaps, receptively. $m \ll M$. When building large-scale maps from a limited amount of data, we have $N \lessapprox M$. For fair comparisons, we set the number of submaps to be equal to the number of local GPs in BCM.

Table 7.2: Computational time of terrain data (in seconds)

| | Training | Prediction | Bayesian fusion | Submapping | Recovery |
|---|---|---|---|---|---|
| **subGPBF** | 43.21 | included in submapping | included in submapping | 10.43+3.26 | / |
| **GMRF-BF** | 11.17 | 0.45 | 0.14 | / | 13.27 |
| **subGMRF-BF** | 11.17 | included in submapping | included in submapping | 3.21+2.67 | 0.65 |
| GPBF | 43.21 | 1.30 | 80.33 | / | / |
| GPBF-BCM | 22.53 | 125.44 | 2.47 | / | / |
| CCIS | 43.21 | included in submapping | included in submapping | 5.59 | / |
| ICIS | 11.17 | included in submapping | included in submapping | 2.96 | 0.63 |
| sparseGPBF | 76.84 | 2.14 | 79.29 | / | / |
| naïveGPBF | / | / | 0.007 | / | / |

The time cost of submapping in subGPBF and subGMRF-BF includes the forward submapping (prediction, fusion, forward update) and the backward update. Multi-threaded parallel computation is implmented for submap recovery.

### 7.2.2 Quantitative Evaluation

Table 7.1 presents computational complexity analysis for each of the evaluated methods. Table 7.2 and Table 7.3 summarise the real computation time, i.e. how long a portion of the program takes to run on the elevation data and the pipe wall thickness data, respectively.

Generally, the global approaches including correlations, i.e. GPBF and GMRF-BF, are more expensive than submap-based approaches. This is because the most expensive computation, interdependently of whether it is in the covariance or the information form, is to factorise the whole $M \times M$ covariance or information matrix. $M$ is often bigger than $10^4$ for a large-scale map. GPBF includes the entire spatial correlations, hereby it is the most time-consuming method. GMRF-BF only models the local neighbourhood structure in the global information matrix, and thus becomes less expensive. When the sparsity of the information matrix (or the correlations between points) is changed, the complexity shown in Table 7.1 remains the same for all columns except for the cost of recovering the mean and variance, as it will increase when data is more correlated. Another observation from the tables is that Bayesian fusion in sparseGPBF costs almost the same

Table 7.3: Computational time of pipes' wall thickness data (in seconds)

| | Training | Prediction | Bayesian fusion | Submapping | Recovery |
|---|---|---|---|---|---|
| **subGPBF** | 8.39 | included in submapping | included in submapping | 4.19+1.74 | / |
| **GMRF-BF** | 4.08 | 0.21 | 0.08 | / | 5.25 |
| **subGMRF-BF** | 4.08 | included in submapping | included in submapping | 1.76+1.45 | 0.35 |
| GPBF | 8.39 | 0.35 | 12.54 | / | / |
| GPBF-BCM | 8.01 | 39.09 | 1.42 | / | / |
| CCIS | 8.39 | included in submapping | included in submapping | 3.17 | / |
| ICIS | 4.08 | included in submapping | included in submapping | 1.62 | 0.34 |
| sparseGPBF | 9.88 | 0.48 | 9.66 | / | / |
| naïveGPBF | / | / | 0.006 | / | / |

The time cost of submapping in subGPBF and subGMRF-BF includes the forward submapping (prediction, fusion, forward update) and the backward update. Multithreaded parallel computation is implmented for submap recovery.

Table 7.4: RMSE± std of terrain data in the 5-run Monte-Carlo simulation (in metre)

| | Prediction | Bayesian fusion | Forward update | Backward update |
|---|---|---|---|---|
| **subGPBF** | 20.25±0.78 | / | 8.09±0.33 | 7.98±0.33 |
| **GMRF-BF** | 24.63±0.81 | 8.05±0.30 | / | / |
| **subGMRF-BF** | 24.63±0.81 | / | 8.32±0.35 | 8.18±0.34 |
| GPBF | 20.25±0.78 | 7.95±0.28 | / | / |
| GPBF-BCM | 25.77±0.96 | 11.84±0.57 | / | / |
| CCIS | 20.25±0.78 | 10.35±0.44 | / | 8.43±0.32 |
| ICIS | 24.63±0.81 | 10.71±0.45 | / | 8.52±0.31 |
| sparseGPBF | 28.17±0.80 | 15.51±0.29 | / | / |
| naïveGPBF | / | 18.05±0.30 | / | / |

Since there is a limited amount of trainig data, we train a GP model and use it in GPBF, subGPBF and CCIS; the same GMRF model is also used in GMRF-BF, subGMRF-BF and ICIS.

time as GPBF. This is based on the fact that the GP inference is in fact a process of conditioning, therefore the sparsity of the covariance matrix will be lost. The naïveGPBF approach costs the least time, yet point measurements are not propagated to the neighbouring points since the correlations between them are not considered.

Table 7.5: RMSE of pipe wall thickness data (in mm)

| | Prediction | Bayesian fusion | Forward submapping | Final map |
|---|---|---|---|---|
| **subGPBF** | 3.38 | / | 2.09 | 2.02 |
| **GMRF-BF** | 3.56 | 2.11 | / | / |
| **subGMRF-BF** | 3.56 | / | 2.22 | 2.13 |
| GPBF | 3.38 | 2.02 | / | / |
| GPBF-BCM | 3.77 | 2.75 | / | / |
| CCIS | 3.38 | 2.54 | / | 2.24 |
| ICIS | 3.56 | 2.61 | / | 2.30 |
| sparseGPBF | 3.94 | 2.82 | / | / |
| naïveGPBF | / | 3.30 | / | / |

Since there is a limited amount of trainig data, we train a GP model and use it in GPBF, subGPBF and CCIS; the same GMRF model is also used in GMRF-BF, subGMRF-BF and ICIS.

In contrast, the submap based methods have a substantial gain in speed with the two global approaches previously mentioned. However, GPBF-BCM is extremely slow during prediction since it computes the prediction on all local GP regressors before getting the weighted averaged output. SubGPBF takes much less time than GPBF since all computations are done within submaps and there is no need to factorising the dense, full covariance matrix. SubGMRF-BF speeds up GMRF-BF by exploiting the GMRF properties and building CI submaps. The total computation time of subGMRF-BF, including the submap recovery, is shorter than that of subGPBF. Even the most expensive computation of recovering local submaps takes very little time when the parallel programming is used. Similarly, ICIS is faster than CCIS.

Table 7.4 and Table 7.5 quantify the results concerning Root Mean Squared Error (RMSE) for all methods. Table 7.4 summarises the average and standard deviation of the absolute RMSE of a 5-runs Monte Carlo Simulation of the terrain data. The three methods proposed in this thesis demonstrate significant advantages over others and achieve comparable RMSE with GPBF. The tables also show clearly that the proposed correlation propagation algorithms step significantly improves the final results. In naïveGPBF, RMSE is computed using only the data that has been updated during fusion.

## 7.3 Summary

Experimental results on a synthetic elevation dataset and a real pipe wall thickness dataset show that when compared with five benchmark approaches, the three approaches proposed as part of this thesis exhibit different levels of considerable computational gain while maintaining competitive accuracy, making them appealing for fast, large-scale 2.5D mapping. Depending on the computational resources and the desired accuracy, one may select methods as follows: If the major concern is speed, one is well advised to use subGMRF-BF. However, this method achieves a slightly worse accuracy than subGPBF and GMRF-BF, and requires an additional step to recover the mean and variance. On the other hand, if accurate predictions are the primary concern, one may expect best results with subGPBF. For less than $10^6$ points, one is well recommended to use GMRF-BF.

# Chapter 8

# Conclusion

This thesis has developed three efficient algorithms for constructing spatially correlated, high-resolution probabilistic 2.5D maps from multiple sources of noisy data. To overcome the severe scalability limitations of GP prediction and Bayesian fusion with correlations, this thesis has investigated how to efficiently and effectively approximate the full spatial correlations with lower computational cost while maintaining accuracy. This chapter briefly reviews our achievements, discusses limitations and draws future plans.

## 8.1 Final Remarks

- **Unified frameworks**: Three unified frameworks for large-scale probabilistic mapping and data fusion have been developed. Instead of completely modelling the correlations, the three frameworks use either local or global approximation instead of the full correlations, thus reducing the memory and computational cost.
  - SubGPBF integrates CI submapping and the novel forward and backward update algorithms between CI submaps with GPBF. Spatial correlations are learned through a GP to generate the prior submaps which later will be corrected using Bayesian fusion or the forward update algorithm. After building all submaps, one step of the backward update can recover the optimal global map.

- GMRF-BF introduces the continuously indexed GMRF into robotic grid mapping and combines it with the information-form Bayesian fusion. The GMRF representation allows the speed-up sparse matrix calculation, and Bayesian fusion only requires a simple addition operation. The combination of them makes GMRF-BF efficient.
- SubGMRF-BF combines GMRF-BF with CI submapping and the novel information-form correlation propagation algorithms between CI submaps. Spatial correlations are learned by a continuous GMRF model to generate the prior submaps which later will be corrected using the information-form Bayesian fusion and the forward update algorithm. After building all submaps, one step of the backward update can recover the optimal global map in linear time. By exploiting the CI property between information-form MVN submaps, the computational complexity is further reduced compared with subGPBF and GMRF-BF.
- For the three frameworks, there is a trade-off between the speed and accuracy, as shown in Table 7.2 to 7.5. Even so, all three methods maintain the high accuracy. Generally speaking, GMRF-BF may be preferred for small datasets since it contains less computational steps. However, for more than $10^6$ points, subGMRF-BF and subGPBF should be applied. Furthermore, subGMRF-BF is suitable when efficiency is the major concern and subGPBF when accuracy is more important.

◇ **Divide-and-conquer**: This thesis takes the divide-and-conquer strategy and develops two CI submapping methods, at the core of which are the correlation propagation algorithms. The covariance-form and information-form propagation algorithms have some similarities, yet the latter possesses several advantages over the former.

Similarities: they both (1) ensure the current submap to be update-to-date, i.e. to summarise all information available; (2) recover the optimal global map given the submap CI property, without referring to the full covariance or information matrix; and (3) allow information to be transmitted and shared through CI submaps both forwards and backwards.

Differences: (1) the submap CI property is naturally depicted in the sparse information matrix in the latter; (2) the latter enables each submap to integrate new information by just adding the new information to the shared parts between CI submaps; while the former need to update the non-common components, which consumes more memory and time.

◇ **Connections between subGPBF and subGMRF-BF**

- Structure: SubGPBF and subGMRF-BF have a parallel structure, except for the additional step of map recovery in the latter. They are equivalent in the correlated Bayesian fusion and the information propagation algorithms.
- Computational and memory cost: Both subGPBF and subGMRF-BF only need to save and compute submaps. subGMRF-BF is more efficient than subGPBF in prior mapping, correlated fusion and correlation propagation, while the map recovery is the most expensive computation in subGMRF-BF. However, experimental results show that subGMRF-BF, even including the recovery step, is faster than subGPBF for the size of maps analysed.

◇ **Explicit link**: This thesis clarifies how the following methodologies relate to each other:

- GP, with the dense covariance matrix,
- GMRF, with the sparse information matrix,
- CI property for continuous mapping in robotics,
- Bayes network.

GP can be seen as a fully connected Bayes network in which each node represents one realisation of the latent function, $f(\mathbf{x}_i)$, and this leads to a dense covariance matrix. Meanwhile, GMRF lets each node only depend on some nearby neighbours, which is equivalent to imposing CI property. The missing edges in the Bayes network correspond to the zeros entries in the information matrix in GMRF. Based on these facts, subGPBF and subGMRF-BF make each node in the Bayes network represent the set of elements in each submap, and the CI property between the nodes is imposed, which results in a sparse information matrix, and the CI property between nodes is imposed. In addition, the direct connections between the GMRF-SPDE model and Matérn GP have been investigated in GMRF-BF. We believe this thesis is a single point reference to studying such relationships in the area of robotic grid mapping, and the results can be applied to other problems.

◇ **Generality**: The three proposed frameworks, including the techniques such as the novel continuous GMRF mapping and correlation propagation algorithms, are generic and applicable as general fusion, mapping, prediction and interpolation methodologies in contexts

where transition models cannot be used, data can be divided into blocks and these blocks can be considered conditionally independent. For instance, we have proved that subGPBF can be easily modified for online sequential submapping with one source of sensor data.

## 8.2 Limitations of the Research

As was summarised in Section 8.1, this thesis has developed global and local techniques for efficient inference and learning of spatially correlated data applied to the problem of 2.5D mapping from large-scale noisy data. However, due to the complexity of the research problems, there remains some challenges as follows.

Currently, subGPBF and subGMRF-BF can only cope with sequences of submaps that form simple Bayes network models, i.e. only two consecutive submaps share a common node. This limitation underlines the difficulty of maintaining CI property when the Bayes network models have become more complicated. For instance, when more than two submaps share one node, more edges need to be added, and the CI property between submaps no longer holds.

This thesis focuses on managing a large amount of query inputs $X^*$ in GP regression, which is an active research issue but previous studies have not attempted to do it for grid maps that contain correlations. To handle a vast amount of training inputs $\{X, \mathbf{y}\}$, the subset approximation methods can be directly applied to subGPBF (see Section 4.4.2). As to GMRF-BF and subGMRF-BF, the continuously indexed GMRF uses the finite element methods and thus can handle a relatively big amount of training data.

In subGMRF-BF, updating the shared part between two submaps requires inverting two blocks of information matrices, e.g. $\Delta_c$ and $\Lambda_a^a$ in Algorithm 6. This is due to the fact that marginalisation of MVNs is more complex in the information form. Nevertheless, factorising such sparse and small dimensional matrices does not affect the efficiency of subGMRF-BF. Another limitation of subGMRF-BF is that GMRF-SPDE directly outputs the information matrix of the basis functions, which needs to be projected to get the information matrix for the submaps. However, due to the sparsity of matrices, the multiplication is efficient.

Finally, this thesis only considers MVNs and assumes that sensor measurements' locations (inputs to GPs) are known and accurate. The two assumptions may not be valid under real circumstances yet are widely used. Another limitation is that experiments in this thesis use at most 20000 points for evaluation. This is because the workstation cannot allocate enough memory for factorising dense covariance matrices bigger than $20000 \times 20000$ in GPBF, the globally optimal benchmark method for comparison. Experiments remain to be tested on larger scale of data ($> 10^6$).

## 8.3 Future Work

There are three vital questions to be answered in the future:

- How to perfect the proposed algorithms? What are their potential applications?
- How to extend the usage of GPs from mapping into the general state estimation problem?
- Can other research topics in the robotics field take advantages of the power of GPs?

A future line of this research lies on the correlation propagation algorithms that do not require the CI property and are suitable for complex graphical models, e.g. submaps may have links with other submaps that are not their neighbours. Piniés et al. (2010) have investigated the backward propagation algorithm when loop closures are present in the map. In addition, how to incorporate the inputs (2D locations) uncertainty into the GP and GMRF inference is worth exploring. The uncertain inputs may need to be modelled with a distribution, which often makes Bayesian inferences more complicated and time-consuming. Some works have been developed in the literature (Kersting et al. 2007; O'Callaghan et al. 2010; Mchutchon and Rasmussen 2011; Jadidi et al. 2017), and they can be incorporated with the research in this thesis. Another challenge in inserting inputs uncertainty into subGPBF and subGMRF-BF is how to propagate the inputs uncertainty between submaps, which, to our knowledge, has not been studied before.

It is straightforward to adapt the proposed methods to online scenarios for assisting robot exploration. SubGPBF and subGMRF-BF naturally suit the online task, as they can handle new incoming data and use the data to update the previous estimates. A more efficient implementation of GMRF-BF is needed, since GMRF-BF factorises the information matrix of a global map.

Some research has been done in this area using GPBF for small maps (Popovic et al. 2017), which could be efficiently adapted for large-scale online mapping using the proposed work proposed. Considering that 3D maps can better represent the environment, we are currently looking into adapting subGPBF and subGMRF-BF for recovering continuous, high-resolution probabilistic 3D surfaces and occupancy maps from a large amount of point clouds, inspired by the work of Gaussian Process Implicit Surfaces (GPIS) (Williams and Fitzgibbon 2007) and Kim and Kim (2014).

Not only for mapping, GPs can be applied for solving the state estimation problems. These problems in mobile robotics are typically addressed by discretising the robot trajectory, and computing the optimal estimate of the robot pose as well as the coordinates of the landmarks in the environments. Such discretisation assumption conflicts with the physical status of the robotic system, in which the robot motion is continuous. To address this problem, Tong et al. (2013) developed a batch estimation method by combining GPs with Gauss-Newton optimisation algorithm. Barfoot et al. (2014) and Anderson et al. (2015) revisited the GP regression problem and the trajectory is viewed as a One-Dimensional (1D) GP with time as the independent variable. By exploiting the Markov property of the prior and the sparsity of the inverse kernel matrix, inferring the continuous trajectory can be achieved efficiently. The issues concerned in continuous state estimation using GPs coincide with the focused problems in this thesis. Therefore, it is worth investigating how to apply the algorithms proposed in the thesis to make the state estimation problems more accurate and efficient.

Finally, GPs are promising for some other robotic applications and have already been successfully used in motion planning and robot manipulation. For example, Mukadam et al. (2016) proposed Gaussian Process Motion Planner (GPMP), which does not rely on a discrete state parametrisation. The continuous-time trajectory is represented as a sample from a GP generated by a Linear Time Varying-Stochastic Differential Equation (LTV-SDE). Combined with gradient based optimisation method, GPMP can find the optimal trajectories given a limited number of states. The methodologies in Dong et al.; Mukadam et al. (2016); Mukadam et al. share many similarities with this thesis, thus exploiting the motion planning field will be one line of future work. In addition, the continuous surface generated by the GPIS algorithm has been used for robot manipulation and object grasping. Given a 3D model described by a GPIS, Mahler et al. (2015) proposed GP-GPIS-OPT, an algorithm for computing grasps of parallel-jaw grippers using Sequential Convex Programming (SCP). This is also one of the interesting fields to explore.

# Appendix A

# Bayesian Fusion for Linear Gaussian Systems

This appendix derives the covariance-form Bayesian fusion (see (3.69) and (3.70)) and its dual form (see (3.75) and (3.76)). The principle idea is to first derive the joint distribution, $p(\mathbf{x}, \mathbf{z}|\mathbf{y}) = p(\mathbf{x}|\mathbf{y})p(\mathbf{z}|\mathbf{x}, \mathbf{y})$, and then to use properties in Section 3.2.2 for computing $p(\mathbf{x}|\mathbf{y}, \mathbf{z})$. More details are as follows.

As (3.68) shows , in the covariance form, the aim is to obtain $p(\mathbf{x}|\mathbf{y}, \mathbf{z}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}^+, \Sigma^+)$ given the prior $p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_\mathbf{x}, \Sigma_\mathbf{x})$ and the likelihood function $p(\mathbf{z}|\mathbf{x}) = \mathcal{N}(H\mathbf{x}, R)$. Based on the fact that $\mathbf{y}$ and $\mathbf{z}$ are conditionally independent given $\mathbf{x}$, we have $p(\mathbf{z}|\mathbf{x}) = p(\mathbf{z}|\mathbf{x}, \mathbf{y})$. Therefore, the log of the joint distribution is as follows (dropping irrelevant constants):

$$\log p(\mathbf{x}, \mathbf{z}|\mathbf{y}) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})^\top \Sigma_\mathbf{x}^{-1}(\mathbf{x} - \boldsymbol{\mu}_\mathbf{x}) - \frac{1}{2}(\mathbf{z} - H\mathbf{x})^\top R^{-1}(\mathbf{z} - H\mathbf{x}) \tag{A.1}$$

It is clear that (A.1) is a joint Gaussian distribution, since it is the exponential of a quadratic form. Expanding out the quadratic terms involving $\mathbf{x}$ and $\mathbf{z}$, and ignoring linear and constant terms,

(A.1) becomes

$$-\frac{1}{2}\mathbf{x}^\top\Sigma_{\mathbf{x}}^{-1}\mathbf{x}-\frac{1}{2}\mathbf{z}^\top R^{-1}\mathbf{z}-\frac{1}{2}(H\mathbf{x})^\top R^{-1}(H\mathbf{x})+\mathbf{z}^\top R^{-1}H\mathbf{x} \tag{A.2}$$

$$=-\frac{1}{2}\begin{pmatrix}\mathbf{x}\\ \mathbf{z}\end{pmatrix}^\top\begin{pmatrix}\Sigma_{\mathbf{x}}^{-1}+H^\top R^{-1}H & -H^\top R^{-1}\\ -R^{-1}H & R^{-1}\end{pmatrix}\begin{pmatrix}\mathbf{x}\\ \mathbf{z}\end{pmatrix} \tag{A.3}$$

$$\triangleq-\frac{1}{2}\begin{pmatrix}\mathbf{x}\\ \mathbf{z}\end{pmatrix}^\top\Sigma^{-1}\begin{pmatrix}\mathbf{x}\\ \mathbf{z}\end{pmatrix}, \tag{A.4}$$

where the information matrix of the joint distribution is (based on the definition of MVN in Section 3.2.2)

$$\Sigma^{-1}=\begin{pmatrix}\Sigma_{\mathbf{x}}^{-1}+H^\top R^{-1}H & -H^\top R^{-1}\\ -R^{-1}H & R^{-1}\end{pmatrix}\triangleq\Lambda=\begin{pmatrix}\Lambda_{\mathbf{xx}} & \Lambda_{\mathbf{xz}}\\ \Lambda_{\mathbf{zx}} & \Lambda_{\mathbf{zz}}\end{pmatrix}. \tag{A.5}$$

Then we can derive $\Sigma^+$ and $\boldsymbol{\mu}^+$ based on the marginalisation property of MVN (see Theorem 3.1 and 3.3), the fact that $\boldsymbol{\mu}_{\mathbf{z}}=H\boldsymbol{\mu}_{\mathbf{x}}$, and the *Woodbury formula* (Woodbury 1949):

$$(A+UCV)^{-1}=A^{-1}-A^{-1}U(C^{-1}+VA^{-1}U)^{-1}VA^{-1}, \tag{A.6}$$

where $A$, $U$, $C$ and $V$ all denote matrices ot the correct size. We get

$$\Sigma^+=\Lambda_{\mathbf{xx}}^{-1}=(\Sigma_{\mathbf{x}}^{-1}+H^\top R^{-1}H)^{-1} \tag{A.7}$$

$$=\Sigma_{\mathbf{x}}-\Sigma_{\mathbf{x}}H^\top(R+H\Sigma_{\mathbf{x}}H^\top)^{-1}H\Sigma_{\mathbf{x}}, \tag{A.8}$$

$$\boldsymbol{\mu}^+=\Sigma^+(\Lambda_{\mathbf{xx}}\boldsymbol{\mu}_{\mathbf{x}}-\Lambda_{\mathbf{xz}}(\mathbf{z}-\boldsymbol{\mu}_{\mathbf{z}})) \tag{A.9}$$

$$=\Sigma^+((\Sigma_{\mathbf{x}}^{-1}+H^\top R^{-1}H)\boldsymbol{\mu}_{\mathbf{x}}+H^\top R^{-1}(\mathbf{z}-H\boldsymbol{\mu}_{\mathbf{x}}) \tag{A.10}$$

$$=\Sigma^+(\Sigma_{\mathbf{x}}^{-1}\boldsymbol{\mu}_{\mathbf{x}}+H^\top R^{-1}\mathbf{z}). \tag{A.11}$$

When substituting (A.8) into (A.11), we obtain

$$\begin{aligned}
\boldsymbol{\mu}^+ &= \boldsymbol{\mu}_{\mathbf{x}} - \Sigma_{\mathbf{x}} H^\top (R + H\Sigma_{\mathbf{x}} H^\top)^{-1} H\boldsymbol{\mu}_{\mathbf{x}} + \Sigma_{\mathbf{x}} H^\top R^{-1}\mathbf{z} - \Sigma_{\mathbf{x}} H^\top (R + H\Sigma_{\mathbf{x}} H^\top)^{-1} H\Sigma_{\mathbf{x}} H^\top R^{-1}\mathbf{z} \\
&= \boldsymbol{\mu}_{\mathbf{x}} - \Sigma_{\mathbf{x}} H^\top (R + H\Sigma_{\mathbf{x}} H^\top)^{-1} H\boldsymbol{\mu}_{\mathbf{x}} + \Sigma_{\mathbf{x}} H^\top (R + H\Sigma_{\mathbf{x}} H^\top)^{-1} ((R + H\Sigma_{\mathbf{x}} H^\top) - H\Sigma_{\mathbf{x}} H^\top) R^{-1}\mathbf{z} \\
&= \boldsymbol{\mu}_{\mathbf{x}} - \Sigma_{\mathbf{x}} H^\top (R + H\Sigma_{\mathbf{x}} H^\top)^{-1} H\boldsymbol{\mu}_{\mathbf{x}} + \Sigma_{\mathbf{x}} H^\top (R + H\Sigma_{\mathbf{x}} H^\top)^{-1} (I + H\Sigma_{\mathbf{x}} H^\top R^{-1} - H\Sigma_{\mathbf{x}} H^\top R^{-1})\mathbf{z} \\
&= \boldsymbol{\mu}_{\mathbf{x}} - \Sigma_{\mathbf{x}} H^\top (R + H\Sigma_{\mathbf{x}} H^\top)^{-1} H\boldsymbol{\mu}_{\mathbf{x}} + \Sigma_{\mathbf{x}} H^\top (R + H\Sigma_{\mathbf{x}} H^\top)^{-1}\mathbf{z} \\
&= \boldsymbol{\mu}_{\mathbf{x}} - \Sigma_{\mathbf{x}} H^\top (R + H\Sigma_{\mathbf{x}} H^\top)^{-1} (\mathbf{z} - H\boldsymbol{\mu}_{\mathbf{x}}).
\end{aligned} \tag{A.12}$$

By now, (3.69) and (3.70) have been proven.

In the information form, we aim to derive $p(\mathbf{x}|\mathbf{y},\mathbf{z}) = \mathcal{N}_c(\mathbf{x}|\boldsymbol{\eta}^+, Q^+)$, given the Gaussian prior $p(\mathbf{x}|\mathbf{y}) = \mathcal{N}_c(\boldsymbol{\eta}_{\mathbf{x}}, Q_{\mathbf{x}})$ and the likelihood function $p(\mathbf{z}|\mathbf{x}) = \mathcal{N}_c(\boldsymbol{\eta}_{\mathbf{z}|\mathbf{x}}, Q_R)$. In Section 3.5.3, the canonical parameters are defined as

$$\boldsymbol{\eta}_{\mathbf{x}} = \Sigma_{\mathbf{x}}^{-1}\boldsymbol{\mu}_{\mathbf{x}}, \quad Q_{\mathbf{x}} = \Sigma_{\mathbf{x}}^{-1}, \quad Q_R = R^{-1}. \tag{A.13}$$

Based on the definition of information vectors, we get

$$\boldsymbol{\eta}^+ = (\Sigma^+)^{-1}\boldsymbol{\mu}^+ \tag{A.14}$$

$$= (\Sigma^+)^{-1}(\Sigma^+(\Sigma_{\mathbf{x}}^{-1}\boldsymbol{\mu}_{\mathbf{x}} + H^\top R^{-1}\mathbf{z})) \tag{A.15}$$

$$= \Sigma_{\mathbf{x}}^{-1}\boldsymbol{\mu}_{\mathbf{x}} + H^\top R^{-1}\mathbf{z}. \tag{A.16}$$

By substituting (A.13) into (A.16), $\boldsymbol{\eta}^+$ becomes

$$\boldsymbol{\eta}^+ = \boldsymbol{\eta}_{\mathbf{x}} + H^\top Q_R\mathbf{z}. \tag{A.17}$$

Based on the definition of information matrices and using (A.7) and (A.13), we have

$$Q^+ = (\Sigma^+)^{-1} = Q_{\mathbf{x}} + H^\top Q_R H. \tag{A.18}$$

By now, (3.75) and (3.76) have been proven.

# Appendix B

# Proof of the Forward Update Algorithm

This appendix proves Corollary 4.2. The goal is to prove that the currently optimal submap $s_2^+ \sim p(\xi_b, \xi_c|\mathbf{z}_a)$, and optionally the currently optimal local map $p(\xi_a, \xi_b, \xi_c|\mathbf{z}_a)$, can be computed from the currently optimal submap $s_1^+ \sim p(\xi_a, \xi_b|\mathbf{z}_a)$ and the prior submap $s_2^- \sim p(\xi_b, \xi_c)$. The only assumption is the submap CI property in (4.7).

Considering the chain rule and the submap CI property that $\mathbf{z}_a$ is conditionally independent of $\xi_c$ given $\xi_b$, we can write the currently optimal submap $s_2^+$ as

$$\begin{aligned} s_2^+ \sim p(\xi_{s_2}|\mathbf{z}_a) &= p(\xi_{s_2}|\xi_b, \mathbf{z}_a) p(\xi_b|\mathbf{z}_a) \\ &= p(\xi_{s_2}|\xi_b) p(\xi_b|\mathbf{z}_a) . \end{aligned} \tag{B.1}$$

Given the prior $p(\xi_{s_2})$ in (4.4), the mean and covariance of $p(\xi_{s_2}|\xi_b)$ is

$$\boldsymbol{\mu}_{s_2|\xi_b} = \boldsymbol{\mu}_{s_2} + \begin{bmatrix} P_b \\ P_{cb} \end{bmatrix} P_b^{-1} (\xi_b - \boldsymbol{\mu}_b) , \quad \text{and} \tag{B.2}$$

$$P_{s_2|\xi_b} = P_{s_2} - \begin{bmatrix} P_b \\ P_{cb} \end{bmatrix} P_b^{-1} \begin{bmatrix} P_b \\ P_{bc} \end{bmatrix}^\top . \tag{B.3}$$

When (B.2) is considered, the mean of $p(\boldsymbol{\xi}_{s_2}|\mathbf{z}_a)$ in (B.1) becomes

$$\begin{aligned}\boldsymbol{\mu}_{s_2}^a = \mathbb{E}\left[\boldsymbol{\xi}_{s_2}|\mathbf{z}_a\right] &= \int d\boldsymbol{\xi}_b \boldsymbol{\mu}_{s_2|\xi_b} p(\boldsymbol{\xi}_b|\mathbf{z}_a) \\ &= \boldsymbol{\mu}_{s_2} + \begin{bmatrix} P_b \\ P_{cb} \end{bmatrix} P_b^{-1}\left(\boldsymbol{\mu}_b^a - \boldsymbol{\mu}_b\right) \\ &= \begin{bmatrix} \boldsymbol{\mu}_b^a \\ \boldsymbol{\mu}_c + P_{cb}P_b^{-1}(\boldsymbol{\mu}_b^a - \boldsymbol{\mu}_b) \end{bmatrix},\end{aligned} \tag{B.4}$$

which shows that the mean of $\boldsymbol{\xi}_b$ in $s_2^+$ is the same as that in the currently optimal submap $s_1^+$, thus $\boldsymbol{\mu}_b^a$ is no longer needed to be computed again while building $s_2^+$.

On the other hand, for computing the covariance of $s_2^+$, i.e. $P_{s_2}^a$, we first combine (B.2) and (B.4), thus getting

$$\boldsymbol{\mu}_{s_2|\xi_b} - \boldsymbol{\mu}_{s_2}^a = \begin{bmatrix} P_b \\ P_{cb} \end{bmatrix} P_b^{-1}\left(\boldsymbol{\xi}_b - \boldsymbol{\mu}_b^a\right), \tag{B.5}$$

and therefore

$$\int d\boldsymbol{\xi}_b p(\boldsymbol{\xi}_b|\mathbf{z}_a)\left(\boldsymbol{\mu}_{s_2|\xi_b} - \boldsymbol{\mu}_{s_2}^a\right)\left(\boldsymbol{\mu}_{s_2|\xi_b} - \boldsymbol{\mu}_{s_2}^a\right)^\top = \begin{bmatrix} P_b \\ P_{cb} \end{bmatrix} P_b^{-1} P_b^a P_b^{-1} \begin{bmatrix} P_b \\ P_{bc} \end{bmatrix}^\top. \tag{B.6}$$

Thus, with reference to (B.3) and (B.6), we obtain

$$\begin{aligned}P_{s_2}^a = \mathbb{COV}[\boldsymbol{\xi}_{s_2}|\mathbf{z}_a] &= \int d\boldsymbol{\xi}_b p(\boldsymbol{\xi}_b|\mathbf{z}_a)\left[P_{s_2|\xi_b} + \left(\boldsymbol{\mu}_{s_2|\xi_b} - \boldsymbol{\mu}_{s_2}^a\right)\left(\boldsymbol{\mu}_{s_2|\xi_b} - \boldsymbol{\mu}_{s_2}^a\right)^\top\right] \\ &= P_{s_2} + \begin{bmatrix} P_b \\ P_{cb} \end{bmatrix} P_b^{-1}\left(P_b^a - P_b\right)P_b^{-1}\begin{bmatrix} P_b \\ P_{bc} \end{bmatrix}^\top \\ &= P_{s_2} + \begin{bmatrix} I \\ P_{cb}P_b^{-1} \end{bmatrix}\left(P_b^a - P_b\right)\begin{bmatrix} I \\ P_b^{-1}P_{bc} \end{bmatrix} \\ &= \begin{bmatrix} P_b & P_{bc} \\ P_{cb} & P_c \end{bmatrix} + \begin{bmatrix} P_b^a - P_b & (P_b^a - P_b)P_b^{-1}P_{bc} \\ P_{cb}P_b^{-1}(P_b^a - P_b) & P_{cb}P_b^{-1}(P_b^a - P_b)P_b^{-1}P_{bc} \end{bmatrix} \\ &= \begin{bmatrix} P_b^a & P_b^a P_b^{-1}P_{bc} \\ P_{cb}P_b^{-1}P_b^a & P_c + P_{cb}P_b^{-1}(P_b^a P_b^{-1}P_{bc} - P_{bc}) \end{bmatrix}.\end{aligned} \tag{B.7}$$

which shows that the covariance of $\boldsymbol{\xi}_b$ in $s_2^+$ is the same as that in $s_1^+$, thus $P_b^a$ is no longer needed to be computed again while building $s_2^+$.

The posterior distribution $p(\boldsymbol{\xi}_c|\mathbf{z}_a)$ can be obtained by marginalising out $\boldsymbol{\xi}_b$ from $p(\boldsymbol{\xi}_{s_2}|\mathbf{z}_a) = \mathcal{N}(\boldsymbol{\mu}_{s_2}^a, P_{s_2}^a)$. Based on Theorem 3.1, the posterior mean and the auto-covariance of $\boldsymbol{\xi}_c$ is

$$\boldsymbol{\mu}_c^a = \boldsymbol{\mu}_c + P_{cb}P_b^{-1}(\boldsymbol{\mu}_b^a - \boldsymbol{\mu}_b), \quad \text{and} \tag{B.8}$$

$$P_c^a = P_c + P_{cb}P_b^{-1}(P_b^a P_b^{-1} P_{bc} - P_{bc}). \tag{B.9}$$

Furthermore, we can get the cross-covariance between $\boldsymbol{\xi}_b$ and $\boldsymbol{\xi}_c$ from (B.7) as

$$P_{cb}^a = P_{cb}P_b^{-1}P_b^a. \tag{B.10}$$

(B.8), (B.9) and (B.10) are used to update the terms related with $\boldsymbol{\xi}_c$ in the local map. We can also define an additional variable $K_f \triangleq P_{cb}P_b^{-1}$ as in Algorithm 2 for simplicity. Thus Corollary 4.2 has been proven. □

# Appendix C

# Proof of the Forward Update Algorithm: An Alternative Way

Alternatively, as described in Piniés and Tardós (2008), the forward update can also be proven in the following way.

Based on the submap CI property, we have

$$p(\boldsymbol{\xi}_c|\boldsymbol{\xi}_b, \mathbf{z}_a) = p(\boldsymbol{\xi}_c|\boldsymbol{\xi}_b). \tag{C.1}$$

The left term $p(\boldsymbol{\xi}_c|\boldsymbol{\xi}_b, \mathbf{z}_a)$ in (C.1) can be obtained by marginalising out $\boldsymbol{\xi}_a$ from $p(\boldsymbol{\xi}_a, \boldsymbol{\xi}_b, \boldsymbol{\xi}_c|\mathbf{z}_a)$ (see (4.6)) and then conditioning on $\boldsymbol{\xi}_b$. The mean and covariance of $p(\boldsymbol{\xi}_c|\boldsymbol{\xi}_b, \mathbf{z}_a)$ are as follows:

$$\boldsymbol{\mu}_{c|b} = \boldsymbol{\mu}_c^a + P_{cb}^a P_b^{-1}(\boldsymbol{\xi}_b - \boldsymbol{\mu}_b^a), \tag{C.2}$$

$$P_{c|b} = P_c^a - P_{cb}^a P_b^{-1} P_{bc}^a. \tag{C.3}$$

Define

$$p(\boldsymbol{\xi}_c|\boldsymbol{\xi}_b) = \mathcal{N}(\boldsymbol{\mu}_{c|b}, P_{c|b}). \tag{C.4}$$

The right term $p(\boldsymbol{\xi}_c|\boldsymbol{\xi}_b)$ in (C.1) can be obtained from the submap $s_2^-$ (see (4.4)) by conditional on $\boldsymbol{\xi}_b$. The mean and covariance of $p(\boldsymbol{\xi}_c|\boldsymbol{\xi}_b)$ are as follows:

$$\boldsymbol{\mu}_{c|b} = \boldsymbol{\mu}_c + P_{cb}P_b^{-1}(\boldsymbol{\xi}_b - \boldsymbol{\mu}_b) \tag{C.5}$$

$$P_{c|b} = P_c - P_{cb}P_b^{-1}P_{bc}\,. \tag{C.6}$$

Equating (C.2), (C.3) and (C.5), (C.6) for all $\boldsymbol{\xi}_b$, and after some manipulations, we obtain the following equations to update the terms related with $\boldsymbol{\xi}_c$:

$$P_{cb}^a = P_{cb}P_b^{-1}P_b^a\,, \tag{C.7}$$

$$\boldsymbol{\mu}_c^a = \boldsymbol{\mu}_c + P_{cb}P_b^{-1}(\boldsymbol{\mu}_b^a - \boldsymbol{\mu}_b)\,, \tag{C.8}$$

$$P_c^a = P_c + P_{cb}P_b^{-1}(P_b^aP_b^{-1}P_{bc} - P_{bc})\,. \tag{C.9}$$

Although unnecessary, the cross-correlation term $P_{ca}^a$ can also be computed. To compute it, we condition $p(\boldsymbol{\xi}_a, \boldsymbol{\xi}_b, \boldsymbol{\xi}_c|\mathbf{z}_a)$ on $\boldsymbol{\xi}_b$, thus getting the covariance of $p(\boldsymbol{\xi}_a, \boldsymbol{\xi}_c|\boldsymbol{\xi}_b, \mathbf{z}_a)$ as

$$\begin{bmatrix} P_a^a - P_{ab}^a(P_b^a)^{-1}P_{ba}^a & P_{ac}^a - P_{ab}^a(P_b^a)^{-1}P_{bc} \\ P_{ca}^a - P_{cb}^a(P_b^a)^{-1}P_{ba}^a & P_c^a - P_{cb}^a(P_b^a)^{-1}P_{bc}^a \end{bmatrix}. \tag{C.10}$$

Then, since $\boldsymbol{\xi}_a$ and $\boldsymbol{\xi}_c$ are conditionally independent given $\boldsymbol{\xi}_b$, the correlation term in (C.10) must be zero, which gives the correlation term

$$P_{ac}^a = P_{ab}^aP_b^{-1}P_{bc}\,. \tag{C.11}$$

Based on the Marginalisation property in Theorem 3.1, the currently optimal submap $s_2^+$ coincides exactly with the last two blocks of $p(\boldsymbol{\xi}_a, \boldsymbol{\xi}_b, \boldsymbol{\xi}_c|\mathbf{z}_a)$. By now Corollary 4.2 has been proven. □

# Appendix D

# Proof of the Backward Update Algorithm

This appendix proves Corollary 4.4. The goal is to prove that the globally optimal submap $s_1^{++} \sim p(\xi_a, \xi_b | \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c)$ can be obtained, once given the globally optimal submap $s_2^{++} \sim p(\xi_b, \xi_c | \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c)$ and the prior submap $s_1^{+} \sim p(\xi_a, \xi_b | \mathbf{z}_a)$. The only assumption is the submap CI property in (4.14).

Using the chain rule and the submap CI property, we have

$$\begin{aligned} s_1^{++} \sim p(\xi_{s_1} | \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c) &= p(\xi_{s_1} | \xi_b, \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c) p(\xi_b | \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c) \\ &= p(\xi_{s_1} | \xi_b, \mathbf{z}_a) p(\xi_b | \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c), \end{aligned} \tag{D.1}$$

where the second equality is based on the CI property that the variable set $(\mathbf{z}_b, \mathbf{z}_c)$ is conditionally independent of $\xi_a$ given the node $\xi_b$.

In (D.1), the term $p(\xi_{s_1} | \xi_b, \mathbf{z}_a)$ represents the pdf of $\xi_{s_1}^a$ (see (4.3)) conditioned on $\xi_b$. Define

$$p(\xi_{s_1} | \xi_b, \mathbf{z}_a) = \mathcal{N}(\boldsymbol{\mu}_{s_1|\xi_b}^a, P_{s_1|\xi_b}^a). \tag{D.2}$$

Furthermore, when using the Conditioning property in Theorem 3.2, we get the mean vector and covariance in (D.2) as follows:

$$\boldsymbol{\mu}^a_{s_1|\xi_b} = \boldsymbol{\mu}^a_{s_1} + \begin{bmatrix} P^a_{ab} \\ P^a_b \end{bmatrix} (P^a_b)^{-1} \left(\boldsymbol{\xi}_b - \boldsymbol{\mu}^a_b\right), \quad \text{and} \tag{D.3}$$

$$P^a_{s_1|\xi_b} = P^a_{s_1} - \begin{bmatrix} P^a_{ab} \\ P^a_b \end{bmatrix} (P^a_b)^{-1} \begin{bmatrix} P^a_{ba} \\ P^a_b \end{bmatrix}^\top. \tag{D.4}$$

Considering (D.3), we can compute the mean of $s_1^{++}$ in (4.12), with reference to (D.3), as

$$\begin{aligned} \boldsymbol{\mu}^{abc}_{s_1} = \mathbb{E}\left[\boldsymbol{\xi}_{s_1}|\mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c\right] &= \int d\boldsymbol{\xi}_b \boldsymbol{\mu}^a_{s_1|\xi_b} p(\boldsymbol{\xi}_b|\mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c) \\ &= \boldsymbol{\mu}^a_{s_1} + \begin{bmatrix} P^a_{ab} \\ P^a_b \end{bmatrix} (P^a_b)^{-1} \left(\boldsymbol{\mu}^{abc}_b - \boldsymbol{\mu}^a_b\right) \\ &= \begin{bmatrix} \boldsymbol{\mu}^a_a + P^a_{ab}(P^a_b)^{-1}(\boldsymbol{\mu}^{abc}_b - \boldsymbol{\mu}^a_b) \\ \boldsymbol{\mu}^{abc}_b \end{bmatrix}, \end{aligned} \tag{D.5}$$

which shows that the mean of $\boldsymbol{\xi}_a$ in $s_1^{++}$ is updated, while the mean of $\boldsymbol{\xi}_b$ in $s_1^{++}$ is the same as that in $s_2^{++}$. Based on the Marginalisation property in Theorem 3.1, the posterior mean of $\boldsymbol{\xi}_a$ can be obtained from (D.5) as

$$\boldsymbol{\mu}^{abc}_a = \boldsymbol{\mu}^a_a + P^a_{ab}(P^a_b)^{-1}(\boldsymbol{\mu}^{abc}_b - \boldsymbol{\mu}^a_b). \tag{D.6}$$

To compute the covariance $s_1^{++}$ in (4.12), i.e. $P^{abc}_{s_1}$, we first combine (D.3) and (D.5), thus getting

$$\boldsymbol{\mu}^a_{s_1|\xi_b} - \boldsymbol{\mu}^{abc}_{s_1} = \begin{bmatrix} P^a_{ab} \\ P^a_b \end{bmatrix} (P^a_b)^{-1} \left(\boldsymbol{\xi}_b - \boldsymbol{\mu}^{abc}_b\right), \tag{D.7}$$

and therefore

$$\int d\boldsymbol{\xi}_b p(\boldsymbol{\xi}_b|\mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c) \left(\boldsymbol{\mu}^a_{s_1|\xi_b} - \boldsymbol{\mu}^{abc}_{s_1}\right)\left(\boldsymbol{\mu}^a_{s_1|\xi_b} - \boldsymbol{\mu}^{abc}_{s_1}\right)^\top = \begin{bmatrix} P^a_{ab} \\ P^a_b \end{bmatrix} (P^a_b)^{-1} P^{abc}_b (P^a_b)^{-1} \begin{bmatrix} P^a_{ba} \\ P^a_b \end{bmatrix}^\top. \tag{D.8}$$

Thus with reference to (D.4) and (D.8), we get

$$
\begin{aligned}
P_{s_1}^{abc} = \mathbb{COV}[\boldsymbol{\xi}_{s_1}|\mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c] &= \int d\boldsymbol{\xi}_b p(\boldsymbol{\xi}_b|\mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c)\left[P_{s_1|\xi_b}^{a} + \left(\boldsymbol{\mu}_{s_1|\xi_b}^{a} - \boldsymbol{\mu}_{s_1}^{abc}\right)\left(\boldsymbol{\mu}_{s_1|\xi_b}^{a} - \boldsymbol{\mu}_{s_1}^{abc}\right)^{\top}\right] \\
&= P_{s_1}^{a} + \begin{bmatrix} P_{ab}^{a} \\ P_{b}^{a} \end{bmatrix} (P_b^a)^{-1}\left(P_b^{abc} - P_b^a\right)(P_b^a)^{-1}\begin{bmatrix} P_{ba}^{a} \\ P_{b}^{a} \end{bmatrix}^{\top} \\
&= \begin{bmatrix} P_a^a & P_{ab}^a \\ P_{ba}^a & P_b^a \end{bmatrix} + \begin{bmatrix} P_{ab}^a(P_b^a)^{-1}(P_b^{abc} - P_b^a)(P_b^a)^{-1}P_{ba}^a & P_{ab}^a(P_b^a)^{-1}(P_b^{abc} - P_b^a) \\ (P_b^{abc} - P_b^a)(P_b^a)^{-1}P_{ba}^a & P_b^{abc} - P_b^a \end{bmatrix} \\
&= \begin{bmatrix} P_a^a + P_{ab}^a(P_b^a)^{-1}(P_b^{abc}(P_b^a)^{-1}P_{ba}^a - P_{ba}^a) & P_{ab}^a(P_b^a)^{-1}P_b^{abc} \\ P_b^{abc}(P_b^a)^{-1}P_a^{ba} & P_b^{abc} \end{bmatrix}.
\end{aligned}
\tag{D.9}
$$

which shows that the covariance of $\boldsymbol{\xi}_a$ in $s_1^{++}$ is updated, while $\boldsymbol{\xi}_b$ in $s_1^{++}$ remains the same as that in $s_2^{++}$. Based on the Marginalisation property of MVN, the auto-covariance of $\boldsymbol{\xi}_a$ and the cross-covariance between $\boldsymbol{\xi}_a$ and $\boldsymbol{\xi}_b$ are obtained by directly extracting the corresponding blocks in (D.9), thus

$$
P_a^{abc} = P_a^a + P_{ab}^a(P_b^a)^{-1}(P_b^{abc}(P_b^a)^{-1}P_{ba}^a - P_{ba}^a), \tag{D.10}
$$

$$
P_{ab}^{abc} = P_{ab}^a(P_b^a)^{-1}P_b^{abc}. \tag{D.11}
$$

When substituting (D.11) in (D.10) and defining $K_b \triangleq P_{ab}^a(P_b^a)^{-1}$, we get $P_a^{abc}$ as it is in Algorithm 3. By now, Corollary 4.4 has been proven. □

# Appendix E

# The Explicit Link between GMRF-SPDEs and Matérn GPs

This appendix introduces some details about the GMRF-SPDE model (Lindgren et al. 2011), and presents the explicit expression for computing the information matrix.

Assuming $f(\mathbf{x})$ is a realisation of GRF $\xi$ situated at $\mathbf{x} \in \mathbb{R}^D$. Considering that $f(\mathbf{x})$ with Matérn covariance is a solution to the linear fractional SPDE, the GMRF representation of $\xi$ is obtained by solving the SPDE

$$(\kappa^2 - \Delta)^{\alpha/2}[\tau f(\mathbf{x})] = \boldsymbol{W}(\mathbf{x}), \quad \kappa > 0, \quad \tau > 0. \tag{E.1}$$

Here $\kappa$ is the spatial scale. $\Delta$ is the Laplacian operator. $\alpha$ controls the smoothness of the realisation. $\tau$ controls the variances. $\boldsymbol{W}$ is the Gaussian white noise. These parameters are explicitly linked to the parameters of Matérn covariance function (see (3.51)) as

$$\sigma_f^2 = \Gamma(\nu)(\Gamma(\alpha)(4\pi)^{D/2}\kappa^{2\nu}\tau^2)^{-1}, \tag{E.2}$$

$$\nu = \alpha - D/2, \tag{E.3}$$

$$l = \sqrt{8\nu}/\kappa. \tag{E.4}$$

In 2.5D mapping, $\mathbf{x} \in \mathbb{R}^2$, thus (E.2), (E.3) and (E.4) become

$$\sigma_f^2 = \Gamma(\nu)(\Gamma(\alpha)4\pi\kappa^{2\nu}\tau^2)^{-1}\,, \tag{E.5}$$

$$\nu = \alpha - 1\,, \tag{E.6}$$

$$l = \sqrt{8\nu}/\kappa\,. \tag{E.7}$$

$\nu$ is usually fixed; and for data located at 2D locations, $\nu$ must be a positive integral to ensure that the Matérn field is Markovian (Lindgren et al. 2011). (E.4) is the empirical definition of $l$. The range parameter $l$ controls the distance at which the correlation decreases almost to zero ($< 0.1$) and decreases fairly slowly despite of $\nu$ (Cressie 1992).

After getting the parameters, there are two ways to compute the information matrix.

**Approach 1**

This approach is only applicable to $\mathbf{x}$ located at uniform 2D grid cells. When $\nu = 0$, the entries in the information matrix for one single point is

$$\begin{array}{ccc} & -1 & \\ -1 & a & -1\,, \\ & -1 & \end{array} \tag{E.8}$$

where $a = \kappa^2 + 4$. Lindgren et al. (2011) derived that the coefficients are computed by convolving (E.8) $\nu$ times. For example, when $\nu = 1$, the information matrix for one single point becomes

$$\begin{array}{ccccc} & & 1 & & \\ & 2 & -2a & 2 & \\ 1 & -2a & a^2+4 & -2a & 1\,. \\ & 2 & -2a & 2 & \\ & & 1 & & \end{array} \tag{E.9}$$

This result is intuitive since $\nu$ is the smoothness parameter, so if the process is smoother, the information matrix has values over a larger neighbourhood. However, it does not imply averaging over larger neighbourhood to make it smoother.

**Approach 2**

The second method can calculate the information matrix for the irregularly located points. The finite element method is used, and the domain is divided into a set of non-intersecting triangles (see Krainski et al. (2017) for details). The approximation is considered as

$$f(\mathbf{x}) = \sum_{k=1}^{m} \psi_k(\mathbf{x}) w_k \,, \tag{E.10}$$

where $\psi_k, k = 1,\ldots,m$ is a basis function, $w_k$ is a Gaussian distributed weight, and $m$ is the number of vertices (the three corners of a triangle) in the triangulation. $\psi_k = 1$ at vertex $k$ and $\psi_k = 0$ at all other vertices. $w_k$ is the value of the field at the vertex $k$. The finite element method can interpolate for any point inside the triangulated domain. Define the matrices $C$, $G$ and $K$ with entries

$$C_{i,j} = \langle \psi_i, \psi_j \rangle, \quad G_{i,j} = \langle \nabla\psi_i, \nabla\psi_j \rangle, \quad K_{i,j} = \kappa^2 C_{i,j} + G_{i,j} \,, \tag{E.11}$$

to get the information matrix $Q$ as a function of $\kappa$ and $\nu$:

$$Q = K \,, \quad \text{when } \nu = 0 \tag{E.12}$$

$$Q = KC^{-1}K \,, \quad \text{when } \nu = 1 \tag{E.13}$$

$$Q = KC^{-1}Q_{\nu-1}C^{-1}K, \quad \text{when } \nu = 2, 3, \ldots \,. \tag{E.14}$$

Again, increasing $\nu$ will make the information matrix denser. The information matrix $Q$ can be generalized for a fractional values of $\nu$ using the Taylor approximation, see the author's discussion in (Lindgren et al. 2011, pp. 493).

# Appendix F

# Proof of the Information-Form Forward Update Algorithm

This appendix proves Corollary 6.1. The goal is to prove that the currently optimal submap $s_2^+ \sim p(\xi_b, \xi_c|\mathbf{z}_a)$, as was defined in (6.4), and optionally the globally optimal local map $p(\xi_a, \xi_b, \xi_c|\mathbf{z}_a)$ (see (6.5)), can be obtained using the forward update algorithm, once given two conditionally independent submaps $s_1^+ \sim p(\xi_a, \xi_b|\mathbf{z}_a)$ (see (6.2)) and $s_2^- \sim p(\xi_b, \xi_c)$ (see (6.3)). The only assumption is the submap CI property in (4.7).

Based on the submap CI property, we have

$$p(\xi_a|\xi_b, \xi_c, z_a) = p(\xi_a|\xi_b, z_a) \tag{F.1}$$

$$p(\xi_c|\xi_b, \xi_a, z_a) = p(\xi_c|\xi_b) \tag{F.2}$$

In (F.1), $p(\xi_a|\xi_b, \xi_c, z_a)$ can be obtained from the local map $p(\xi_a, \xi_b, \xi_c|\mathbf{z}_a)$ by first marginalising out $\xi_c$, and then conditioning on $\xi_b$. First, we marginalise out $\xi_c$ and get $p(\xi_a, \xi_b|z_a)$, which can be denoted as

$$p(\xi_a, \xi_b|z_a) = \mathcal{N}_c(\beta_{ab}^{G,a}, \Lambda_{ab}^{G,a}), \tag{F.3}$$

where the superscript $^{G,a}$ denotes the estimate comes from $p(\boldsymbol{\xi}_a, \boldsymbol{\xi}_b, \boldsymbol{\xi}_c|\mathbf{z}_a)$. Furthermore, by using Theorem 3.3, we get

$$\begin{aligned}\boldsymbol{\beta}_{ab}^{G,a} &= \begin{bmatrix}\boldsymbol{\eta}_a^a \\ \boldsymbol{\eta}_b^a\end{bmatrix} - \begin{bmatrix}0 \\ Q_{bc}^a\end{bmatrix}(Q_c^a)^{-1}\boldsymbol{\eta}_c^a \\ &= \begin{bmatrix}\boldsymbol{\eta}_a^a \\ \boldsymbol{\eta}_b^a - Q_{bc}^a(Q_c^a)^{-1}\boldsymbol{\eta}_c^a\end{bmatrix}\end{aligned}, \quad \text{and} \tag{F.4}$$

$$\begin{aligned}\Lambda_{ab}^{G,a} &= \begin{bmatrix}Q_a^a & Q_{ab}^a \\ Q_{ba}^a & Q_b^a\end{bmatrix} - \begin{bmatrix}0 \\ Q_{bc}^a\end{bmatrix}(Q_c^a)^{-1}\begin{bmatrix}0 & Q_{cb}^a\end{bmatrix} \\ &= \begin{bmatrix}Q_a^a & Q_{ab}^a \\ Q_{ba}^a & Q_b^a - Q_{bc}^a(Q_c^a)^{-1}Q_{cb}^a\end{bmatrix}.\end{aligned} \tag{F.5}$$

Then, we condition (F.3) on $\boldsymbol{\xi}_b$ using Theorem 3.4, and get

$$p(\boldsymbol{\xi}_a|\boldsymbol{\xi}_b, z_a) = \mathcal{N}_c(\boldsymbol{\eta}_a^a - Q_{ab}^a\boldsymbol{\xi}_b, Q_a^a). \tag{F.6}$$

On the In (F.1), $p(\boldsymbol{\xi}_a|\boldsymbol{\xi}_b, z_a)$ can be obtained from submap $s_1^+$ by conditioning on $\boldsymbol{\xi}_b$, thus

$$p(\boldsymbol{\xi}_a|\boldsymbol{\xi}_b, z_a) = \mathcal{N}_c(\boldsymbol{\beta}_a^a - \Lambda_{ab}^a\boldsymbol{\xi}_b, \Lambda_a^a)\,, \tag{F.7}$$

Since (F.6) equals to (F.7) according to (F.1), we have

$$\boldsymbol{\eta}_a^a - Q_{ab}^a\boldsymbol{\xi}_b = \boldsymbol{\beta}_a^a - \Lambda_{ab}^a\boldsymbol{\xi}_b\,, \tag{F.8}$$

$$Q_a^a = \Lambda_a^a\,, \tag{F.9}$$

thus

$$Q_{ab}^a = \Lambda_{ab}^a\,, \tag{F.10}$$

$$\boldsymbol{\eta}_a^a = \boldsymbol{\beta}_a^a. \tag{F.11}$$

On the other hand, the left and right term in (F.2) can be computed in a similar way as above. To compute the left term in (F.2), we first calculate $p(\xi_b, \xi_c|z_a)$ by marginalising out $\xi_a$ from $p(\xi_a, \xi_b, \xi_c|\mathbf{z}_a)$, thus we get

$$p(\xi_b, \xi_c|z_a) = \mathcal{N}_c(\zeta_{bc}^{G,a}, \Delta_{bc}^{G,a}), \quad \text{where} \tag{F.12}$$

$$\begin{aligned}\zeta_{bc}^{G,a} &= \begin{bmatrix}\eta_b^a\\ \eta_c^a\end{bmatrix} - \begin{bmatrix}Q_{ba}^a\\ 0\end{bmatrix}(Q_a^a)^{-1}\eta_a^a \\ &= \begin{bmatrix}\eta_b^a - Q_{ba}^a(Q_a^a)^{-1}\eta_a^a\\ \eta_c^a\end{bmatrix}\end{aligned}, \tag{F.13}$$

$$\begin{aligned}\Delta_{bc}^{G,a} &= \begin{bmatrix}Q_b^a & Q_{bc}^a\\ Q_{cb}^a & Q_c^a\end{bmatrix} - \begin{bmatrix}Q_{ba}^a\\ 0\end{bmatrix}(Q_a^a)^{-1}\begin{bmatrix}Q_{ab}^a & 0\end{bmatrix} \\ &= \begin{bmatrix}Q_b^a - Q_{ba}^a(Q_a^a)^{-1}Q_{ab}^a & Q_{bc}^a\\ Q_{cb}^a & Q_c^a\end{bmatrix}\end{aligned}. \tag{F.14}$$

Then, we condition (F.12) on $\xi_b$, and get

$$p(\xi_c|\xi_b, z_a) = \mathcal{N}_c(\eta_c^a - Q_{cb}^a\xi_b, Q_c^a). \tag{F.15}$$

As to the right term in (F.2), we can get $p(\xi_c|\xi_b)$ by conditioning $s_2^-$ on $\xi_b$, thus we get

$$p(\xi_c|\xi_b) = \mathcal{N}_c(\zeta_c - \Delta_{cb}\xi_b, \Delta_c). \tag{F.16}$$

Since (F.15) is equal to (F.16) based on (F.2), we obtain

$$\eta_c^a - Q_{cb}^a\xi_b = \zeta_c - \Delta_{cb}\xi_b, \tag{F.17}$$

$$Q_c^a = \Delta_c. \tag{F.18}$$

Therefore

$$\eta_c^a = \zeta_c, \tag{F.19}$$

$$Q_{cb}^a = \Delta_{cb}. \tag{F.20}$$

Since both (6.2) and (F.3) represent $s_1^+$, we obtain

$$\begin{bmatrix} \beta_a^a \\ \beta_b^a \end{bmatrix} = \begin{bmatrix} \eta_a^a \\ \eta_b^a - Q_{bc}^a (Q_c^a)^{-1} \eta_c^a \end{bmatrix}, \quad \text{and} \tag{F.21}$$

$$\begin{bmatrix} \Lambda_a^a & \Lambda_{ab}^a \\ \Lambda_{ba}^a & \Lambda_b^a \end{bmatrix} = \begin{bmatrix} Q_a^a & Q_{ab}^a \\ Q_{ba}^a & Q_b^a - Q_{bc}^a (Q_c^a)^{-1} Q_{cb}^a \end{bmatrix}. \tag{F.22}$$

Thus by comparing each entry in (F.21) and (F.22), we get

$$\eta_b^a = \beta_b^a + Q_{bc}^a (Q_c^a)^{-1} \eta_c^a , \tag{F.23}$$

$$Q_b^a = \Lambda_b^a + Q_{bc}^a (Q_c^a)^{-1} Q_{cb}^a , \tag{F.24}$$

and by substituting (F.18), (F.19) and (F.20) in (F.23) and (F.24), we get:

$$\eta_b^a = \beta_b^a + \Delta_{bc} \Delta_c^{-1} \zeta_c , \tag{F.25}$$

$$Q_b^a = \Lambda_b^a + \Delta_{bc} \Delta_c^{-1} \Delta_{cb} . \tag{F.26}$$

Based on the above deduction, each term in (6.5) can be obtained using the following equations:

$$\eta_a^a = \beta_a^a , \tag{F.27}$$

$$\eta_b^a = \beta_b^a + \Delta_{bc} \Delta_c^{-1} \zeta_c , \tag{F.28}$$

$$\eta_c^a = \zeta_c , \tag{F.29}$$

$$Q_a^a = \Lambda_a^a , \tag{F.30}$$

$$Q_{ab}^a = \Lambda_{ab}^a , \tag{F.31}$$

$$Q_b^a = \Lambda_b^a + \Delta_{bc} \Delta_c^{-1} \Delta_{cb} , \tag{F.32}$$

$$Q_{bc}^a = \Delta_{bc} , \tag{F.33}$$

$$Q_c^a = \Delta_c . \tag{F.34}$$

Therefore the currently optimal local map can be represented as

$$p(\boldsymbol{\xi}_a, \boldsymbol{\xi}_b, \boldsymbol{\xi}_c | \mathbf{z}_a) = \mathcal{N}_c \left( \begin{bmatrix} \boldsymbol{\beta}_a^a \\ \boldsymbol{\beta}_b^a + \Delta_{bc}\Delta_c^{-1}\boldsymbol{\zeta}_c \\ \boldsymbol{\zeta}_c \end{bmatrix}, \begin{bmatrix} \Lambda_a^a & \Lambda_{ab}^a & 0 \\ \Lambda_{ba}^a & \Lambda_b^a + \Delta_{bc}\Delta_c^{-1}\Delta_{cb} & \Delta_{bc} \\ 0 & \Delta_{cb} & \Delta_c \end{bmatrix} \right). \tag{F.35}$$

The currently optimal second submap $s_2^+ \sim p(\boldsymbol{\xi}_b, \boldsymbol{\xi}_c | \mathbf{z}_a)$ can be obtained by marginalising out $\boldsymbol{\xi}_a$ from the local map in (F.35) using Theorem 3.3. Thus we get

$$s_2^+ \sim \mathcal{N}_c \left( \begin{bmatrix} \boldsymbol{\beta}_b^a + \Delta_{bc}\Delta_c^{-1}\boldsymbol{\eta}_c - \Lambda_{ba}^a(\Lambda_a^a)^{-1}\boldsymbol{\beta}_a^a \\ \boldsymbol{\zeta}_c \end{bmatrix}, \begin{bmatrix} \Lambda_b^a + \Delta_{bc}\Delta_c^{-1}\Delta_{cb} - \Lambda_{ba}^a(\Lambda_a^a)^{-1}\Lambda_{ab}^a & \Delta_{bc} \\ \Delta_{cb} & \Delta_c \end{bmatrix} \right). \tag{F.36}$$

(F.36) indicates that only the information vector and matrix of $\boldsymbol{\xi}_b$ in $s_2^-$ need to be corrected during the forward update; and these estimates will include the knowledge from both $s_1^+$ and $s_2^-$ after the forward update.

Finally, Corollary 6.1 has been proven. □

# Appendix G

# Proof of the Information-Form Backward Update Algorithm

This appendix proves Corollary 6.2. The goal is to prove that the globally optimal submap $s_1^{++}$, as was defined in (6.9), and optionally the globally optimal local map $p(\xi_a, \xi_b, \xi_c|\mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c)$ (defined (6.10)), can be obtained using the backward update algorithm, once given the two conditionally independent submaps $s_1^+ \sim p(\xi_a, \xi_b|\mathbf{z}_a)$ (see (6.2)) and $s_2^{++} \sim p(\xi_b, \xi_c|\mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c)$ (see (6.8)). The only assumption is the CI property in (4.14).

Based on submap CI property, we get

$$p(\xi_a|\xi_b, \xi_c, \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c) = p(\xi_a|\xi_b, \mathbf{z}_a), \tag{G.1}$$

$$p(\xi_c|\xi_a, \xi_b, \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c) = p(\xi_c|\xi_b, \mathbf{z}_b, \mathbf{z}_c). \tag{G.2}$$

In (G.1), the left term can be obtained from the local map $p(\xi_a\xi_b, \xi_c|\mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c)$ by first marginalising out $\xi_c$ and then conditioning on $\xi_b$. First, we marginalise out $\xi_c$ and get $p(\xi_a, \xi_b|\mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c)$,

which can be denoted as

$$p(\boldsymbol{\xi}_a, \boldsymbol{\xi}_b | \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c) = \mathcal{N}_c(\boldsymbol{\beta}_{ab}^{G,abc}, \boldsymbol{\Lambda}_{ab}^{G,abc}), \quad \text{where} \tag{G.3}$$

$$\begin{aligned} \boldsymbol{\beta}_{ab}^{G,abc} &= \begin{bmatrix} \boldsymbol{\eta}_a^{abc} \\ \boldsymbol{\eta}_b^{abc} \end{bmatrix} - \begin{bmatrix} 0 \\ Q_{bc}^{abc} \end{bmatrix} (Q_c^{abc})^{-1} \boldsymbol{\eta}_c^{abc} \\ &= \begin{bmatrix} \boldsymbol{\eta}_a^{abc} \\ \boldsymbol{\eta}_b^{abc} - Q_{bc}^{abc} (Q_c^{abc})^{-1} \boldsymbol{\eta}_c^{abc} \end{bmatrix}, \end{aligned} \tag{G.4}$$

$$\begin{aligned} \boldsymbol{\Lambda}_{ab}^{G,abc} &= \begin{bmatrix} Q_a^{abc} & Q_{ab}^{abc} \\ Q_{ba}^{abc} & Q_b^{abc} \end{bmatrix} - \begin{bmatrix} 0 \\ Q_{bc}^{abc} \end{bmatrix} (Q_c^{abc})^{-1} \begin{bmatrix} 0 & Q_{cb}^{abc} \end{bmatrix} \\ &= \begin{bmatrix} Q_a^{abc} & Q_{ab}^{abc} \\ Q_{ba}^{abc} & Q_b^{abc} - Q_{bc}^{abc} (Q_c^{abc})^{-1} Q_{cb}^{abc} \end{bmatrix}. \end{aligned} \tag{G.5}$$

Then, we condition (G.3) on $\boldsymbol{\xi}_b$ and get

$$p(\boldsymbol{\xi}_a | \boldsymbol{\xi}_b, \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c) = \mathcal{N}_c(\boldsymbol{\eta}_a^{abc} - Q_{ab}^{abc} \boldsymbol{\xi}_b, Q_a^{abc}). \tag{G.6}$$

Meanwhile, the right term $p(\boldsymbol{\xi}_a | \boldsymbol{\xi}_b, \mathbf{z}_a)$ in (G.1) can be obtained from the first submap (6.2) by conditioning on $\boldsymbol{\xi}_b$

$$p(\boldsymbol{\xi}_a | \boldsymbol{\xi}_b, \mathbf{z}_a) = \mathcal{N}_c(\boldsymbol{\beta}_a^a - \boldsymbol{\Lambda}_{ab}^a \boldsymbol{\xi}_b, \boldsymbol{\Lambda}_a^a). \tag{G.7}$$

According to CI property (G.1), (G.6) equals to (G.7), thus

$$\boldsymbol{\eta}_a^{abc} - Q_{ab}^{abc} \boldsymbol{\xi}_b = \boldsymbol{\beta}_a^a - \boldsymbol{\Lambda}_{ab}^a \boldsymbol{\xi}_b, \tag{G.8}$$

$$Q_a^{abc} = \boldsymbol{\Lambda}_a^a. \tag{G.9}$$

Meanwhile, since $\boldsymbol{\xi}_a$ and $\mathbf{z}_b, \mathbf{z}_c$ are conditionally independent, we have

$$\boldsymbol{\eta}_a^{abc} = \boldsymbol{\beta}_a^a. \tag{G.10}$$

By substituting (G.10) into (G.8), we get:

$$Q_{ab}^{abc} = \boldsymbol{\Lambda}_{ab}^a. \tag{G.11}$$

In (G.2), the left term can be obtained from the local map by first marginalising out $\boldsymbol{\xi}_a$ and then conditioning on $\boldsymbol{\xi}_b$. We first marginalise out $\boldsymbol{\xi}_a$ and get $p(\boldsymbol{\xi}_b, \boldsymbol{\xi}_c | \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c)$, which can be represented as

$$p(\boldsymbol{\xi}_b, \boldsymbol{\xi}_c | \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c) = \mathcal{N}_c(\boldsymbol{\zeta}_{bc}^{G,abc}, \Delta_{bc}^{G,abc}), \quad \text{where} \tag{G.12}$$

$$\begin{aligned} \boldsymbol{\zeta}_{bc}^{G,abc} &= \begin{bmatrix} \boldsymbol{\eta}_b^{abc} \\ \boldsymbol{\eta}_c^{abc} \end{bmatrix} - \begin{bmatrix} Q_{ba}^{abc} \\ 0 \end{bmatrix} (Q_a^{abc})^{-1} \boldsymbol{\eta}_a^{abc} \\ &= \begin{bmatrix} \boldsymbol{\eta}_b^{abc} - Q_{ba}^{abc} (Q_a^{abc})^{-1} \boldsymbol{\eta}_a^{abc} \\ \boldsymbol{\eta}_c^{abc} \end{bmatrix} \end{aligned}, \tag{G.13}$$

$$\begin{aligned} \Delta_{bc}^{G,abc} &= \begin{bmatrix} Q_b^{abc} & Q_{bc}^{abc} \\ Q_{cb}^{abc} & Q_c^{abc} \end{bmatrix} - \begin{bmatrix} Q_{ba}^{abc} \\ 0 \end{bmatrix} (Q_a^{abc})^{-1} \begin{bmatrix} Q_{ab}^{abc} & 0 \end{bmatrix} \\ &= \begin{bmatrix} Q_b^{abc} - Q_{ba}^{abc} (Q_a^{abc})^{-1} Q_{ab}^{abc} & Q_{bc}^{abc} \\ Q_{cb}^{abc} & Q_c^{abc} \end{bmatrix} \end{aligned}. \tag{G.14}$$

When (G.12) is conditioned on $\boldsymbol{\xi}_b$,

$$p(\boldsymbol{\xi}_c | \boldsymbol{\xi}_b, \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c) = \mathcal{N}_c(\boldsymbol{\eta}_c^{abc} - Q_{cb}^{abc} \boldsymbol{\xi}_b, Q_c^{abc}). \tag{G.15}$$

Meanwhile, the right term in (G.2) could be computed from $s_2^{++}$ in (6.8) by conditioning on $\boldsymbol{\xi}_b$ that

$$p(\boldsymbol{\xi}_c | \boldsymbol{\xi}_b, \mathbf{z}_b, \mathbf{z}_c) = \mathcal{N}_c(\boldsymbol{\zeta}_c^{abc} - \Delta_{cb}^{abc} \boldsymbol{\xi}_b, \Delta_c^{abc}). \tag{G.16}$$

According to CI property (G.2), (G.15) is equal to (G.16), thus

$$\boldsymbol{\eta}_c^{abc} - Q_{cb}^{abc} \boldsymbol{\xi}_b = \boldsymbol{\zeta}_c^{abc} - \Delta_{cb}^{abc} \boldsymbol{\xi}_b, \tag{G.17}$$

$$Q_c^{abc} = \Delta_c^{abc}. \tag{G.18}$$

Since both (G.12) and (6.8) represent the globally optimal second submap, we obtain

$$\begin{bmatrix} \zeta_b^{abc} \\ \zeta_c^{abc} \end{bmatrix} = \begin{bmatrix} \eta_b^{abc} - Q_{ba}^{abc}(Q_a^{abc})^{-1}\eta_a^{abc} \\ \eta_c^{abc} \end{bmatrix}, \tag{G.19}$$

$$\begin{bmatrix} \Delta_b^{abc} & \Delta_{bc}^{abc} \\ \Delta_{cb}^{abc} & \Delta_c^{abc} \end{bmatrix} = \begin{bmatrix} Q_b^{abc} - Q_{ba}^{abc}(Q_a^{abc})^{-1}Q_{ab}^{abc} & Q_{bc}^{abc} \\ Q_{cb}^{abc} & Q_c^{abc} \end{bmatrix}. \tag{G.20}$$

Then by comparing each entries in (G.19) and (G.20), we get

$$\eta_b^{abc} = \zeta_b^{abc} + Q_{ba}^{abc}(Q_a^{abc})^{-1}\eta_a^{abc}, \tag{G.21}$$

$$Q_b^{abc} = \Delta_b^{abc} + Q_{ba}^{abc}(Q_a^{abc})^{-1}Q_{ab}^{abc}. \tag{G.22}$$

When substituting (G.9), (G.10) and (G.11) into (G.21) and (G.22), we get

$$\eta_b^{abc} = \zeta_b^{abc} + \Lambda_{ba}^{a}(\Lambda_a^a)^{-1}\beta_a^a, \tag{G.23}$$

$$Q_b^{abc} = \Delta_b^{abc} + \Lambda_{ba}^{a}(\Lambda_a^a)^{-1}\Lambda_{ab}^{a}. \tag{G.24}$$

In conclusion, given two conditionally independent submaps, i.e. the previous submap and its following submap that is optimal in the sense that it contains all the currently available observations, the optimal global map, represented with canonical parameters, can be computed as

$$\eta_a^{abc} = \beta_a^a, \tag{G.25}$$

$$\eta_b^{abc} = \zeta_b^{abc} + \Lambda_{ba}^{a}(\Lambda_a^a)^{-1}\beta_a^a, \tag{G.26}$$

$$\eta_c^{abc} = \zeta_c^{abc}, \tag{G.27}$$

$$Q_a^{abc} = \Lambda_a^a, \tag{G.28}$$

$$Q_{ab}^{abc} = \Lambda_{ab}^{a}, \tag{G.29}$$

$$Q_b^{abc} = \Delta_b^{abc} + \Lambda_{ba}^{a}(\Lambda_a^a)^{-1}\Lambda_{ab}^{a}, \tag{G.30}$$

$$Q_{bc}^{abc} = \Delta_{bc}^{abc}, \tag{G.31}$$

$$Q_c^{abc} = \Delta_c^{abc}. \tag{G.32}$$

Therefore the currently optimal local map can be represented as

$$p(\boldsymbol{\xi}_a, \boldsymbol{\xi}_b, \boldsymbol{\xi}_c | \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c) = \mathcal{N}_c \left( \begin{bmatrix} \boldsymbol{\beta}_a^a \\ \boldsymbol{\zeta}_b^{abc} + \boldsymbol{\Lambda}_{ba}^a (\boldsymbol{\Lambda}_a^a)^{-1} \boldsymbol{\beta}_a^a \\ \boldsymbol{\zeta}_c^{abc} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Lambda}_a^a & \boldsymbol{\Lambda}_{ab}^a & 0 \\ \boldsymbol{\Lambda}_{ba}^a & \boldsymbol{\Delta}_b^{abc} + \boldsymbol{\Lambda}_{ba}^a (\boldsymbol{\Lambda}_a^a)^{-1} \boldsymbol{\Lambda}_{ab}^a & \boldsymbol{\Delta}_{bc}^{abc} \\ 0 & \boldsymbol{\Delta}_{cb}^{abc} & \boldsymbol{\Delta}_c^{abc} \end{bmatrix} \right). \tag{G.33}$$

The currently optimal submap $s_1^{++} \sim p(\boldsymbol{\xi}_a, \boldsymbol{\xi}_b | \mathbf{z}_a, \mathbf{z}_b, \mathbf{z}_c)$ can be obtained by marginalising out $\boldsymbol{\xi}_c$ from the globally optimal local map in (G.33) using Theorem 3.3. Thus we get its information vector $\boldsymbol{\beta}_{s_1}^{abc}$ and information matrix $\boldsymbol{\Lambda}_{s_1}^{abc}$ as follows:

$$\boldsymbol{\beta}_{s_1}^{abc} = \begin{bmatrix} \boldsymbol{\beta}_a^a \\ \boldsymbol{\zeta}_b^{abc} + \boldsymbol{\Lambda}_{ba}^a (\boldsymbol{\Lambda}_a^a)^{-1} \boldsymbol{\beta}_a^a - \boldsymbol{\Delta}_{bc}^{abc} (\boldsymbol{\Delta}_c^{abc})^{-1} \boldsymbol{\zeta}_c^{abc} \end{bmatrix}, \tag{G.34}$$

$$\boldsymbol{\Lambda}_{s_1}^{abc} = \begin{bmatrix} \boldsymbol{\Lambda}_a^a & \boldsymbol{\Lambda}_{ab}^a \\ \boldsymbol{\Lambda}_{ba}^a & \boldsymbol{\Delta}_b^{abc} + \boldsymbol{\Lambda}_{ba}^a (\boldsymbol{\Lambda}_a^a)^{-1} \boldsymbol{\Lambda}_{ab}^a - \boldsymbol{\Delta}_{bc}^{abc} (\boldsymbol{\Delta}_c^{abc})^{-1} \boldsymbol{\Delta}_{cb}^{abc} \end{bmatrix}. \tag{G.35}$$

The proof of Corollary 6.2 is done. □

# Appendix H

# Application of SubGPBF to Mapping with One Large Data Set

This appendix deduces Algorithm 4, which adapts subGPBF to sequential mapping with one sensing modality, and proves that the optimal global map can be obtained with only the backward update algorithm. We use three submaps $s_1$, $s_2$ and $s_3$ from the graphical model in Figure 4.4a to explain Algorithm 4. For better understanding, three submaps are used instead of two. The components in each submap are defined as

$$\boldsymbol{\xi}_{s_1} = \begin{bmatrix} \boldsymbol{\xi}_a \\ \boldsymbol{\xi}_b \end{bmatrix}, \quad \boldsymbol{\xi}_{s_2} = \begin{bmatrix} \boldsymbol{\xi}_b \\ \boldsymbol{\xi}_c \end{bmatrix}, \quad \boldsymbol{\xi}_{s_3} = \begin{bmatrix} \boldsymbol{\xi}_c \\ \boldsymbol{\xi}_d \end{bmatrix}. \tag{H.1}$$

First consider the forward mapping procedure in Algorithm 4.

**Procedure** 1 **Forward Process**

Denote the GP predicted submap $s_1$ and $s_2$ as

$$s_1^- \sim p(\boldsymbol{\xi}_{s_1}) = \mathcal{N}\left( \begin{bmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{bmatrix}, \begin{bmatrix} P_a & P_{ab} \\ P_{ba} & P_b \end{bmatrix} \right), \tag{H.2}$$

$$s_2^- \sim p(\boldsymbol{\xi}_{s_2}) = \mathcal{N}\left( \begin{bmatrix} \boldsymbol{\mu}_b^{s_2} \\ \boldsymbol{\mu}_c \end{bmatrix}, \begin{bmatrix} P_b^{s_2} & P_{bc} \\ P_{cb} & P_c \end{bmatrix} \right), \tag{H.3}$$

where the superscript $^{s_2}$ represents the estimate of $\boldsymbol{\xi}_b$ in $s_2^-$. Note that from now on the superscript denotes which submap has the component been marginalised from.

Then, define the new observation for the prior submap $s_2^-$ to be $\boldsymbol{\xi}_{b_1}^{s_1^-} \triangleq \boldsymbol{\xi}_b^{s_1^-}$, where the subscript of $_{b_1}$ denotes the first common element between submaps, and the superscript of $^{s_1^-}$ represents the value obtained via marginalisation from $s_1^-$. Similar notation is used throughout this appendix and will not be explained for brevity. When fusing $\boldsymbol{\xi}_{b_1}^{s_1^-}$ with $s_2^-$ via the correlated Bayesian fusion, we get $s_2^+$:

$$s_2^+ \sim p(\boldsymbol{\xi}_{s_2}|\boldsymbol{\xi}_{b_1}^{s_1^-}) = p(\boldsymbol{\xi}_b^{s_2}, \boldsymbol{\xi}_c|\boldsymbol{\xi}_{b_1}^{s_1^-}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_b^{s_2,+} \\ \boldsymbol{\mu}_c^+ \end{bmatrix}, \begin{bmatrix} P_b^{s_2,+} & P_{bc}^+ \\ P_{cb}^+ & P_c^+ \end{bmatrix}\right). \tag{H.4}$$

In (H.4), the mean vector and the covariance matrix is computed using the Bayesian fusion equations in (3.69) and (3.70), respectively.

Next, define the new information for updating GP predicted prior submap $s_3^-$ (see (H.5)) to be $\boldsymbol{\xi}_{b_2}^{s_2^+} \triangleq \boldsymbol{\xi}_c^{s_2^+}$, where $\boldsymbol{\xi}_c^{s_2^+}$ is computed via marginalisation from $s_2^+$.

$$s_3^- \sim p(\boldsymbol{\xi}_{s_3}) = p(\boldsymbol{\xi}_c^{s_3}, \boldsymbol{\xi}_d) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_c^{s_3} \\ \boldsymbol{\mu}_d \end{bmatrix}, \begin{bmatrix} P_c^{s_3} & P_{cd} \\ P_{dc} & P_d \end{bmatrix}\right). \tag{H.5}$$

When fusing the new information $\boldsymbol{\xi}_{b_2}^{s_2^+}$ with $s_3^-$, we get $s_3^+$ :

$$s_3^+ \sim p(\boldsymbol{\xi}_{s_3}|\boldsymbol{\xi}_{b_1}^{s_1^-}, \boldsymbol{\xi}_{b_2}^{s_2^+}) = p(\boldsymbol{\xi}_c^{s_3}, \boldsymbol{\xi}_d|\boldsymbol{\xi}_{b_1}^{s_1^-}, \boldsymbol{\xi}_{b_2}^{s_2^+}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_c^{s_3,+} \\ \boldsymbol{\mu}_d^+ \end{bmatrix}, \begin{bmatrix} P_c^{s_3,+} & P_{cd}^+ \\ P_{dc}^+ & P_d^+ \end{bmatrix}\right). \tag{H.6}$$

Now the forward submapping procedure has ended and we obtain the submap $s_1^-$, $s_2^+$ and $s_3^+$. $s_3^+$ is globally optimal since it contains all the current information.

Note that in (H.4) and (H.6), the new observation, which is used for submap fusion, is different with that in subGPBF. In subGPBF, the new observations are from another dataset $\Psi_2$, and they can be transmitted to the next submap via the forward update algorithm. However, with only one dataset, it is the GP prediction of the common elements that is regarded as the new observation. For instance, since the local training subset is used to train the local GP for each submap, the pdf

of $\xi_b^{s_1^-}$ and $s_2^-$ (see (H.3)) can be written as:

$$\xi_b^{s_1^-} \sim p(\xi_b | X_{s_1}, \mathbf{y}_{s_1}, X_{s_1}^*), \tag{H.7}$$

$$s_2^- \sim p(\xi_{s_2} | X_{s_2}, \mathbf{y}_{s_2}, X_{s_2}^*), \tag{H.8}$$

where $X_{s_1}$ and $\mathbf{y}_{s_1}$ denote the training data and $X_{s_1}^*$ the query points of the local GP for $s_1$, respectively; and $X_{s_2}$, $\mathbf{y}_{s_2}$, $X_{s_1}^*$ are defined in the similar way. It shows in (H.7) and (H.8) that there is no information that is used twice in $\xi_b^{s_1^-}$ and $s_2^-$, thus fusing $\xi_b^{s_1^-}$ and $s_2^-$ is reasonable. And the information contained in $s_1^-$ is transmitted to $s_2^-$ after fusion.

**Procedure** 2 **Backward Update**

After the forward submapping, $s_3^+$ is optimal while others not. To get the globally optimal submap $s_1^+$ and $s_2^{++}$, we now use the backward update algorithm.

Denote $s_2^{++}$ as

$$s_2^{++} \sim p(\xi_{s_2} | \xi_{b_1}^{s_1^-}, \xi_{b_2}^{s_2^+}) = p(\xi_b^{s_2}, \xi_c | \xi_{b_1}^{s_1^-}, \xi_{b_2}^{s_2^+}) = \mathcal{N}\left( \begin{bmatrix} \boldsymbol{\mu}_b^{s_2,++} \\ \boldsymbol{\mu}_c^{++} \end{bmatrix}, \begin{bmatrix} P_b^{s_2,++} & P_{bc}^{++} \\ P_{cb}^{++} & P_c^{++} \end{bmatrix} \right). \tag{H.9}$$

Based on Corollary 4.4, computing the globally optimal $s_2^{++}$ equals to recovering the globally optimal local map, defined as (H.10); and only the terms related with $\xi_b$ need to be updated.

$$p(\xi_b, \xi_c, \xi_d | \xi_{b_1}^{s_1^-}, \xi_{b_2}^{s_2^+}) = \mathcal{N}\left( \begin{bmatrix} \boldsymbol{\mu}_b^{s_2,++} \\ \boldsymbol{\mu}_c^{++} \\ \boldsymbol{\mu}_d^{++} \end{bmatrix}, \begin{bmatrix} P_b^{s_2,++} & P_{bc}^{++} & P_{bd}^{++} \\ P_{cb}^{++} & P_c^{++} & P_{cd}^{++} \\ P_{db}^{++} & P_{dc}^{++} & P_d^{++} \end{bmatrix} \right). \tag{H.10}$$

Furthermore, (H.10) can be written as

$$\begin{aligned} p(\xi_b, \xi_c, \xi_d | \xi_{b_1}^{s_1^-}, \xi_{b_2}^{s_2^+}) &= p(\xi_b | \xi_c, \xi_d, \xi_{b_1}^{s_1^-}, \xi_{b_2}^{s_2^+}) p(\xi_c, \xi_d | \xi_{b_1}^{s_1^-}, \xi_{b_2}^{s_2^+}) \\ &= p(\xi_b | \xi_c, \xi_{b_1}^{s_1^-}, \xi_{b_2}^{s_2^+}) p(\xi_c, \xi_d | \xi_{b_1}^{s_1^-}, \xi_{b_2}^{s_2^+}) \\ &= p(\xi_b | \xi_c, \xi_{b_1}^{s_1^-}) p(\xi_c, \xi_d | \xi_{b_1}^{s_1^-}, \xi_{b_2}^{s_2^+}), \end{aligned} \tag{H.11}$$

where the first equality is based on the chain rule, and the second equality comes from the submap CI property that $\xi_b$ is conditionally independent of $\xi_d$ given $\xi_c$. The third equality is due to that

$\boldsymbol{\xi}_{b_2}^{s_2^+}$ is already contained in $s_2^+$ and thus there is no need to use it to update $s_2^+$.

Then, the mean and covariance of $s_2^{++}$ (see (H.9)) can be computed by directly applying the results of backward update algorithm (see Algorithm 3); and with reference to (H.4) and (H.6), we can get

$$\boldsymbol{\mu}_b^{s_2,++} = \boldsymbol{\mu}_b^{s_2,+} + P_{bc}^+ (P_c^+)^{-1} (\boldsymbol{\mu}_c^{s_3,+} - \boldsymbol{\mu}_c^+), \tag{H.12}$$

$$P_{bc}^{++} = P_{bc}^+ (P_c^+)^{-1} P_c^{s_3,+}, \tag{H.13}$$

$$\boldsymbol{\mu}_c^{++} = \boldsymbol{\mu}_c^{s_3,+}, \tag{H.14}$$

$$P_b^{s_2,++} = P_b^{s_2,+} + P_{bc}^+ (P_c^+)^{-1} (P_{cb}^{++} - P_{cb}^+), \tag{H.15}$$

$$P_c^{++} = P_c^{s_3,+}, \tag{H.16}$$

$$(optionally) P_{bd}^{++} = P_{bc}^+ (P_c^+)^{-1} P_{cd}^+. \tag{H.17}$$

Next, we correct the first submap $s_1^-$ to get the globally optimal submap $s_1^+$, which is denoted as

$$s_1^+ \sim p(\xi_{s_1} | \xi_{b_1}^{s_1^-}, \xi_{b_2}^{s_2^+}) = p(\xi_a, \xi_b | \xi_{b_1}^{s_1^-}, \xi_{b_2}^{s_2^+}) = \mathcal{N}\left( \begin{bmatrix} \boldsymbol{\mu}_a^+ \\ \boldsymbol{\mu}_b^+ \end{bmatrix}, \begin{bmatrix} P_a^+ & P_{ab}^+ \\ P_{ba}^+ & P_b^+ \end{bmatrix} \right). \tag{H.18}$$

Based on Corollary 4.4, computing $s_1^+$ equals to recovering the globally optimal local map:

$$p(\xi_a, \xi_b, \xi_c | \xi_{b_1}^{s_1^-}, \xi_{b_2}^{s_2^+}) = \mathcal{N}\left( \begin{bmatrix} \boldsymbol{\mu}_a^+ \\ \boldsymbol{\mu}_b^+ \\ \boldsymbol{\mu}_c^{++} \end{bmatrix}, \begin{bmatrix} P_a^+ & P_{ab}^+ & P_{ac}^+ \\ P_{ba}^+ & P_b^{++} & P_{bc}^{++} \\ P_{ca}^+ & P_{cb}^{++} & P_c^{++} \end{bmatrix} \right), \tag{H.19}$$

in which only the terms related with $\boldsymbol{\xi}_a$ need to be updated. Furthermore, based on the chain rule, (H.19) can be written as

$$\begin{aligned} p(\xi_a, \xi_b, \xi_c | \xi_{b_1}^{s_1^-}, \xi_{b_2}^{s_2^+}) &= p(\xi_a | \xi_b, \xi_c, \xi_{b_1}^{s_1^-}, \xi_{b_2}^{s_2^+}) p(\xi_b, \xi_c | \xi_{b_1}^{s_1^-}, \xi_{b_2}^{s_2^+}) \\ &= p(\xi_a | \xi_b, \xi_{b_1}^{s_1^-}) p(\xi_b, \xi_c | \xi_{b_1}^{s_1^-}, \xi_{b_2}^{s_2^+}) \\ &= p(\xi_a | \xi_b) p(\xi_b, \xi_c | \xi_{b_1}^{s_1^-}, \xi_{b_2}^{s_2^+}). \end{aligned} \tag{H.20}$$

The second equality in (H.20) comes from the submap CI property that $\boldsymbol{\xi}_a$ is conditionally independent of $\boldsymbol{\xi}_c$ given $\boldsymbol{\xi}_b$. The third equality is based on the fact that $\boldsymbol{\xi}_{b_1}^{s_1^-}$ comes from $s_1^-$ thus there

is no need to use it to correct $s_1^-$.

Then we can directly apply backward update algorithm, and with reference to (H.2) and (H.9), the globally optimal submap $s_1^+$ in (H.18) can be obtained as

$$\boldsymbol{\mu}_a^+ = \boldsymbol{\mu}_a + P_{ab}^+ P_b^{+,-1} (\boldsymbol{\mu}_b^{s_2,++} - \boldsymbol{\mu}_b), \tag{H.21}$$

$$P_{ab}^+ = P_{ab} P_b^{-1} P_b^{s_2,++}, \tag{H.22}$$

$$\boldsymbol{\mu}_b^+ = \boldsymbol{\mu}_b^{s_2,++}, \tag{H.23}$$

$$P_a^+ = P_b + P_{ab} P_b^{-1} (P_{ba}^+ - P_{ba}), \tag{H.24}$$

$$P_b^+ = P_b^{s_2,+}, \tag{H.25}$$

$$(optionally) P_{ac}^+ = P_{ab} P_b^{-1} P_{bc}^{++}. \tag{H.26}$$

By now, the three submaps are globally optimal.

In general, Algorithm 4 can be used to build a sequential of submaps and the optimal global map can be obtained after the backward update. □

# Bibliography


Anderson, S., Barfoot, T. D., Tong, C. H., and Särkkä, S. (2015). Batch nonlinear continuous-time trajectory estimation as exactly sparse gaussian process regression. *Autonomous Robots*, 39(3):221–238.

Bailey, T. and Durrant-Whyte, H. (2006). Simultaneous localization and mapping (slam): Part ii. *IEEE Robotics & Automation Magazine*, 13(3):108–117.

Banerjee, S. and Fuentes, M. (2012). Bayesian modeling for large spatial datasets. *Wiley interdisciplinary reviews. Computational statistics*, 4(1):59–66.

Banerjee, S., Gelfand, A. E., Finley, A. O., and Sang, H. (2008). Gaussian predictive process models for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(4):825–848.

Barfoot, T. D., Tong, C. H., and Särkkä, S. (2014). Batch continuous-time trajectory estimation as exactly sparse gaussian process regression. In *Robotics: Science and Systems*.

Bertram, M., Tricoche, X., and Hagen, H. (2003). Adaptive smooth scattered-data approximation for large-scale terrain visualization. In *VisSym*, pages 177–184.

Bolin, D. and Lindgren, F. (2013). A comparison between markov approximations and other methods for large spatial data sets. *Computational Statistics & Data Analysis*, 61(0):7–21.

Bolin, D. and Wallin, J. (2016). Spatially adaptive covariance tapering. *Spatial Statistics*, 18:163–178.

Borenstein, J., Feng, L., and Everett, H. (1996). *Navigating mobile robots: Systems and techniques*. AK Peters, Ltd.

Bosse, M., Newman, P., Leonard, J., Soika, M., Feiten, W., and Teller, S. (2003). An atlas framework for scalable mapping. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 2, pages 1899–1906. IEEE.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Castanedo, F. (2013). A review of data fusion techniques. *The Scientific World Journal*, 2013.

Chalupka, K., Williams, C. K., and Murray, I. (2013). A framework for evaluating approximation methods for gaussian process regression. *The Journal of Machine Learning Research*, 14(1):333–350.

Chen, Y., Davis, T. A., Hager, W. W., and Rajamanickam, S. (2008). Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. *ACM Transactions on Mathematical Software (TOMS)*, 35(3):22.

Cressie, N. (1992). Statistics for spatial data. *Terra Nova*, 4(5):613–617.

Cuthill, E. and McKee, J. (1969). Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 1969 24th national conference*, pages 157–172. ACM.

Davis, T. A. (2005). Algorithm 849: A concise sparse cholesky factorization package. *ACM Trans. Math. Softw.*, 31(4):587–591.

Dempster, A. P. (1967). Upper and lower probabilities induced by a multivalued mapping. *The annals of mathematical statistics*, pages 325–339.

Diggle, P. J., Ribeiro Jr, P. J., and Christensen, O. F. (2003). *An introduction to model-based geostatistics*, pages 43–86. Springer.

Dissanayake, M. G., Newman, P., Clark, S., Durrant-Whyte, H. F., and Csorba, M. (2001). A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on robotics and automation*, 17(3):229–241.

Dong, J., Mukadam, M., Dellaert, F., and Boots, B. Motion planning as probabilistic inference using gaussian processes and factor graphs.

Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110.

Durrant-Whyte, H. and Henderson, T. (2008). *Multisensor Data Fusion*, book section 26, pages 585–610. Springer Berlin Heidelberg.

Durrant-Whyte, H. and Stevens, M. (2001). Data fusion in decentralised sensing networks. In *4th International Conference on Information Fusion*.

Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57.

Furrer, R., Genton, M. G., and Nychka, D. (2006). Covariance tapering for interpolation of large spatial datasets. *Journal of Computational and Graphical Statistics*, 15(3):502–523.

Gal, Y., van der Wilk, M., and Rasmussen, C. E. (2014). Distributed variational inference in sparse gaussian process regression and latent variable models. In *Advances in Neural Information Processing Systems*, pages 3257–3265.

George, A. (1973). Nested dissection of a regular finite element mesh. *SIAM Journal on Numerical Analysis*, 10(2):345–363.

Gerardo-Castro, M. P., Peynot, T., Ramos, F., and Fitch, R. (2015). Non-parametric consistency test for multiple-sensing-modality data fusion. In *18th International Conference on Information Fusion*, pages 443–451. IEEE.

Girolami, M. (2016). Bayesian data fusion with gaussian process priors: an application to protein fold recognition. In *Workshop on Probabilistic Modeling and Machine Learning in Structural and Systems Biology, PMSB*.

Golub, G. H. and Plemmons, R. J. (1980). Large-scale geodetic least-squares adjustment by dissection and orthogonal decomposition. *Linear Algebra and Its Applications*, 34:3–28.

Golub, G. H. and Van Loan, C. F. (1996). Matrix computations. 1996. *Johns Hopkins University, Press, Baltimore, MD, USA*, pages 374–426.

Grisetti, G., Kummerle, R., Stachniss, C., and Burgard, W. (2010). A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43.

Gutmann, J.-S., Fukuchi, M., and Fujita, M. (2005). A floor and obstacle height map for 3d navigation of a humanoid robot. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 1066–1071. IEEE.

Hartman, L. W. (2006). Bayesian modelling of spatial data using markov random fields, with application to elemental composition of forest soil. *Mathematical Geology*, 38(2):113–133.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI2013)*.

Hodges, J. S. (2013). *Spatial Models as Mixed Linear Models*, book section 5, pages 129–150. CRC Press.

Horn, R. A. and Johnson, C. R. (2012). *Matrix analysis*. Cambridge university press.

Householder, A. S. (1964). *The Theory of Matrices in Numerical Analysis*. Blaisdell, New York.

Hrafnkelsson, B. and Cressie, N. (2003). Hierarchical modeling of count data with application to nuclear fall-out. *Environmental and Ecological Statistics*, 10(2):179–200.

Huang, G. P., Mourikis, A. I., and Roumeliotis, S. I. (2008a). Analysis and improvement of the consistency of extended kalman filter based slam. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 473–479. IEEE.

Huang, G. P., Mourikis, A. I., and Roumeliotis, S. I. (2009). On the complexity and consistency of ukf-based slam. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 4401–4408. IEEE.

Huang, S. and Dissanayake, G. (2007). Convergence and consistency analysis for extended kalman filter based slam. *IEEE Transactions on robotics*, 23(5):1036–1049.

Huang, S., Wang, Z., and Dissanayake, G. (2008b). Sparse local submap joining filter for building large-scale maps. *IEEE Transactions on Robotics*, 24(5):1121–1130.

Jadidi, M. G., Miro, J. V., and Dissanayake, G. (2015). Mutual information-based exploration on continuous occupancy maps. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 6086–6092. IEEE.

Jadidi, M. G., Miro, J. V., and Dissanayake, G. (2017). Warped gaussian processes occupancy mapping with uncertain inputs. *IEEE Robotics and Automation Letters*, 2(2):680–687.

Jadidi, M. G., Miró, J. V., Valencia, R., and Andrade-Cetto, J. (2014). Exploration on continuous gaussian process frontier maps. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6077–6082. IEEE.

Jadidi, M. G., Valls Miró, J., Valencia Carreño, R., Andrade-Cetto, J., and Dissanayake, G. (2013). Exploration using an information-based reaction-diffusion process. In *Proceedings of the 2013Australasian Conference on Robotics & Automation*, pages 1–10.

Julier, S. J. and Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422.

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., and Dellaert, F. (2012). isam2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, 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.

Kassem, M., Shehata, O. M., and Morgan, E. I. (2016). Multi-modal mobile sensor data fusion for autonomous robot mapping problem. In *MATEC Web of Conferences*, volume 42. EDP Sciences.

Kelly, A. and Stentz, A. (1997). Analysis of requirements for high speed rough terrain autonomous mobility. part ii: Resolution and accuracy. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3326–3333.

Kersting, K., Plagemann, C., Pfaff, P., and Burgard, W. (2007). Most likely heteroscedastic gaussian process regression. In *Proceedings of the 24th international conference on Machine learning*, pages 393–400. ACM.

Kidner, D., Dorey, M., and Smith, D. (1999). What's the point? interpolation and extrapolation with a regular grid dem. In *Fourth International Conference on GeoComputation, Fredericksburg, VA, USA*.

Kielbasinski, A. (1987). A note on rounding-error analysis of cholesky factorization. 88-89:487–494.

Kim, S. and Kim, J. (2013). Continuous occupancy maps using overlapping local gaussian processes. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4709–4714.

Kim, S. and Kim, J. (2014). Recursive bayesian updates for occupancy mapping and surface reconstruction. In *Australasian Conference on Robotics and Automation*.

Kim, S., Kim, J., et al. (2011). Towards large-scale occupancy map building using dirichlet and gaussian processes. In *Proceedings of the Australasian Conference on Robotics and Automation*, pages 4756–4761.

Kjærgaard, M., Bayramoglu, E., Massaro, A. S., and Jensen, K. (2011). Terrain mapping and obstacle detection using gaussian processes. In *Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on*, volume 1, pages 118–123. IEEE.

Koller, D. and Friedman, N. (2009). *Gaussian Network Models*. MIT press.

Kortenkamp, D., Bonasso, R., and Murphy, R. (1998). Ai-based mobile robots: Case studies of successful robot systems.

Krainski, E., Lindgren, F., Simpson, D., and Rue, H. (2017). The R-INLA tutorial: SPDE models. *Journal of Geographical Systems, http://www. math. ntnu. no/inla/r-inla. org/tutorials/spde/spde-tutorial.pdf*.

Lacroix, S., Mallet, A., Bonnafous, D., Bauzil, G., Fleury, S., Herrb, M., and Chatila, R. (2002). Autonomous rover navigation on unknown terrains: Functions and integration. *The International Journal of Robotics Research*, 21(10-11):917–942.

Lang, T., Plagemann, C., and Burgard, W. (2007). Adaptive non-stationary kernel regression for terrain modeling. In *Robotics: Science and Systems*, volume 6.

Leonard, J. J. and Feder, H. J. S. (2000). A computationally efficient method for large-scale concurrent mapping and localization. In *Robotics Research*, pages 169–176. Springer.

Li, J. and Heap, A. D. (2014). Spatial interpolation methods applied in the environmental sciences: A review. *Environmental Modelling & Software*, 53:173–189.

Lindgren, F. and Rue, H. (2015). Bayesian spatial modelling with r-inla. *Journal of Statistical Software*, 63(19).

Lindgren, F., Rue, H., and Lindström, J. (2011). An explicit link between gaussian fields and gaussian markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498.

MacDonald, B., Ranjan, P., and Chipman, H. (2013). GPfit: an R package for gaussian process model fitting using a new optimization algorithm. *arXiv preprint arXiv:1305.0759*.

Mahler, J., Patil, S., Kehoe, B., Van Den Berg, J., Ciocarlie, M., Abbeel, P., and Goldberg, K. (2015). Gp-gpis-opt: Grasp planning with shape uncertainty using gaussian process implicit surfaces and sequential convex programming. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 4919–4926. IEEE.

Mahler, R. P. (2007). *Statistical multisource-multitarget information fusion*. Artech House, Inc.

Martino, S. and Rue, H. (2010). Implementing approximate bayesian inference using integrated nested laplace approximation: A manual for the inla program. *Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway. Compiled on April*, 8:2010.

Mchutchon, A. and Rasmussen, C. E. (2011). Gaussian process training with input noise. In *Advances in Neural Information Processing Systems 24*, pages 1341–1349.

Melkumyan, A. and Ramos, F. (2009). A sparse covariance function for exact gaussian process inference in large datasets. In *IJCAI*, volume 9, pages 1936–1942.

Minka, T. P. (2001). *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology.

Mitchell, H. B. (2007). *Multi-sensor data fusion: an introduction*. Springer Science & Business Media.

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2002). Fastslam: A factored solution to the simultaneous localization and mapping problem. In *In Proceedings of the AAAI National Conference on Artificial Intelligence*.

Moore, D. and Russell, S. J. (2015). Gaussian process random fields. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3357–3365.

Mukadam, M., Dong, J., Dellaert, F., and Boots, B. Simultaneous trajectory estimation and planning via probabilistic inference.

Mukadam, M., Yan, X., and Boots, B. (2016). Gaussian process motion planning. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 9–15. IEEE.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Nabney, I. (2002). *NETLAB: algorithms for pattern recognition*. Springer Science & Business Media.

Natural Resources Canada. Laval, Quebec (1995). National Topographic Database, version 3.10. Ottawa: Natural Resources Canada, Geomatics Canada.

Neal, R. M. (1996). *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media.

Neumann, M., Marthaler, D., Huang, s., and Kersting, K. (2014). pyGPs; a library for Gaussian process regression and classification.

Newcombe, R. A. and Davison, A. J. (2010). Live dense reconstruction with a single moving camera. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1498–1505. IEEE.

Nychka, D., Hammerling, D., Sain, S., and Lenssen, N. (2013). Latticekrig: multiresolution kriging based on markov random fields. *URL http://cran. r-project. org/web/packages/LatticeKrig*.

O'Callaghan, S. T. and Ramos, F. T. (2012). Gaussian process occupancy maps. *The International Journal of Robotics Research*, 31(1):42–62.

O'Callaghan, S. T., Ramos, F. T., and Durrant-Whyte, H. (2010). Contextual occupancy maps incorporating sensor and location uncertainty. In *2010 IEEE International Conference on Robotics and Automation*, pages 3478–3485.

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

Orbanz, P. and Teh, Y. (2010). *Bayesian Nonparametric Models.*, pages 81–89. Springer.

Paz, L. M., Piniés, P., Tardós, J. D., and Neira, J. (2008a). Large-scale 6-dof slam with stereo-in-hand. *IEEE transactions on robotics*, 24(5):946–957.

Paz, L. M., Tardós, J. D., and Neira, J. (2008b). Divide and conquer: Ekf slam in $o(n)$. *IEEE Transactions on Robotics*, 24(5):1107–1120.

Pfaff, P., Triebel, R., and Burgard, W. (2007). An efficient extension to elevation maps for outdoor terrain mapping and loop closing. *The International Journal of Robotics Research*, 26(2):217–230.

Piniés, P., Paz, L. M., Gálvez-López, D., and Tardós, J. D. (2010). CI-Graph simultaneous localization and mapping for three-dimensional reconstruction of large and complex environments using a multicamera system. *Journal of Field Robotics*, 27(5):561–586.

Piniés, P., Paz, L. M., and Tardós, J. D. (2009). Ci-graph: An efficient approach for large scale slam. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3913–3920. IEEE.

Piniés, P. and Tardós, J. D. (2008). Large-Scale SLAM building conditionally independent local maps: Application to monocular vision. *IEEE Transactions on Robotics*, 24(5):1094–1106.

Plagemann, C., Kersting, K., and Burgard, W. (2008a). Nonstationary gaussian process regression using point estimates of local smoothness. *Machine learning and knowledge discovery in databases*, pages 204–219.

Plagemann, C., Mischke, S., Prentice, S., Kersting, K., Roy, N., and Burgard, W. (2008b). Learning predictive terrain models for legged robot locomotion. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3545–3552. IEEE.

Plagemann, C., Mischke, S., Prentice, S., Kersting, K., Roy, N., and Burgard, W. (2009). A bayesian regression approach to terrain mapping and an application to legged robot locomotion. *Journal of Field Robotics*, 26(10):789–811.

Popovic, M., Vidal-Calleja, T., Hitz, G., Sa, I., Siegwart, R., and Nieto, J. (2017). Multiresolution mapping and informative path planning for uav-based terrain monitoring. *arXiv preprint arXiv:1703.02854*.

Qiu, Q., Yang, T., and Han, J. (2009). A new real-time algorithm for off-road terrain estimation using laser data. *Science in China Series F: Information Sciences*, 52(9):1658–1667.

Quadrianto, N., Kersting, K., and Xu, Z. (2010). *Gaussian Process*, book section 428-439. Springer US, Boston, MA.

Quiñonero-Candela, J. and Rasmussen, C. E. (2005). A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959.

Ramos, F. and Ott, L. (2015). Hilbert maps: scalable continuous occupancy mapping with stochastic gradient descent. In *Proceedings of Robotics: Science and Systems*.

Rasmussen, C. E. and Nickisch, H. (2010). Gaussian processes for machine learning (gpml) toolbox. *Journal of Machine Learning Research*, 11(Nov):3011–3015.

Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian processes for machine learning*. MIT Press, Cambridge, Mass.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Rue, H. and Follestad, T. (2001). Gmrflib: a c-library for fast and exact simulation of gaussian markov random fields. Technical report, SIS-2002-236.

Rue, H. and Held, L. (2005). *Gaussian Markov random fields: theory and applications*. CRC Press.

Rue, H., Martino, S., and Chopin, N. (2009). Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations. *Journal of the royal statistical society: Series b (statistical methodology)*, 71(2):319–392.

Rue, H., Riebler, A., Sørbye, S. H., Illian, J. B., Simpson, D. P., and Lindgren, F. K. (2017). Bayesian computing with inla: a review. *Annual Review of Statistics and Its Application*, 4:395–421.

Rue, H. and Tjelmeland, H. (2002). Fitting gaussian markov random fields to gaussian fields. *Scandinavian Journal of Statistics*, 29(1):31–49.

Rusu, R. B., Marton, Z. C., Blodow, N., Dolha, M., and Beetz, M. (2008). Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941.

Saad, Y. (2003). *Iterative methods for sparse linear systems*. SIAM.

Schabenberger, O. and Gotway, C. A. (2004). *Statistical methods for spatial data analysis*. CRC press.

Schwaighofer, A. (2005). Bayesian committee machine MATLAB software. https://github.com/sods/bcm.

Schwaighofer, A. and Tresp, V. (2003). Transductive and inductive methods for approximate gaussian process regression. In Obermayer, S. B., Thrun, S., and K., editors, *Advances in Neural Information Processing Systems 15*, pages 977–984.

Shi, L., Miro, J. V., Zhang, T., Vidal-Calleja, T., Sun, L., and Dissanayake, G. (2016). Constrained sampling of 2.5d probabilistic maps for augmented inference. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3131–3136.

Shi, L., Sun, L., Vidal Calleja, T., and Miro, J. V. (2015). Kernel-specific gaussian process for predicting pipe wall thickness maps. In *Australasian Conference on Robotics and Automation 2015*. AARA.

Simpson, D., Lindgren, F., and Rue, H. (2012). In order to make spatial statistics computationally feasible, we need to forget about the covariance function. *Environmetrics*, 23(1):65–74.

Skinner, B., Vidal Calleja, T., Valls Miro, J., De Bruijn, F., and Falque, R. (2014). 3D point cloud upsampling for accurate reconstruction of dense 2.5 d thickness maps. In *Australasian Conference on Robotics and Automation*.

Snelson, E. and Ghahramani, Z. (2006). Sparse gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1257–1264.

Stein, M. L. (1999). *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media.

Sun, L., Vidal-Calleja, T., and Valls Miro, J. (2015). Bayesian fusion using conditionally independent submaps for high reolution 2.5d mapping. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3394–3400.

Sun, L., Vidal-Calleja, T., and Valls Miro, J. (2016). Gaussian markov random fields for fusion in information form. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1840–1845.

Sun, L., Vidal-Calleja, T., and Valls Miro, J. (2017). Coupling conditionally independent submaps for large-scale 2.5d mapping with gaussian markov random fields. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3133–3137.

Sun, Y., Li, B., and Genton, M. G. (2012). *Geostatistics for large datasets*, pages 55–77. Springer.

Tardós, J. D., Neira, J., Newman, P. M., and Leonard, J. J. (2002). Robust mapping and localization in indoor environments using sonar data. *The International Journal of Robotics Research*, 21(4):311–330.

Thompson, P., Nettleton, E., and Durrant-Whyte, H. (2011). Distributed large scale terrain mapping for mining and autonomous systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4236–4241.

Thrun, S. (2003). Learning occupancy grid maps with forward sensor models. *Autonomous robots*, 15(2):111–127.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic robotics*. Cambridge, Mass. : MIT Press.

Thrun, S., Liu, Y., Koller, D., Ng, A. Y., Ghahramani, Z., and Durrant-Whyte, H. (2004). Simultaneous localization and mapping with sparse extended information filters. *The International Journal of Robotics Research*, 23(7-8):693–716.

Tobler, W. R. (1970). A computer movie simulating urban growth in the detroit region. *Economic geography*, 46(sup1):234–240.

Tong, C. H., Furgale, P., and Barfoot, T. D. (2013). Gaussian process gauss–newton for non-parametric simultaneous localization and mapping. *The International Journal of Robotics Research*, 32(5):507–525.

Tresp, V. (2000). A bayesian committee machine. *Neural Computation*, 12(11):2719–2741.

Triebel, R., Pfaff, P., and Burgard, W. (2006). Multi-level surface maps for outdoor terrain mapping and loop closing. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2276–2282. IEEE.

Tse, R., Ahmed, N., and Campbell, M. (2012). Unified mixture-model based terrain estimation with markov random fields. In *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on*, pages 238–243. IEEE.

Tse, R., Ahmed, N. R., and Campbell, M. (2015). Unified terrain mapping model with markov random fields. *IEEE Transactions on Robotics*, 31(2):290–306.

Ulapane, N., Alempijevic, A., Vidal-Calleja, T., Miro, J. V., Rudd, J., and Roubal, M. (2014). Gaussian process for interpreting pulsed eddy current signals for ferromagnetic pipe profiling. In *Proceedings of 9th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 1762–1767, in press.

Vanhatalo, J., Riihimäki, J., Hartikainen, J., Jylänki, P., Tolvanen, V., and Vehtari, A. (2013). Gpstuff: Bayesian modeling with gaussian processes. *Journal of Machine Learning Research*, 14(1):1175–1179.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag New York.

Vasudevan, S. (2012). Data fusion with gaussian processes. *Robotics and Autonomous Systems*, 60(12):1528–1544.

Vasudevan, S., Ramos, F., Nettleton, E., and Durrant-Whyte, H. (2009). Gaussian process modeling of large-scale terrain. *Journal of Field Robotics*, 26(10):812–840.

Vasudevan, S., Ramos, F., Nettleton, E., and Durrant-Whyte, H. (2011). Non-stationary dependent gaussian processes for data fusion in large-scale terrain modeling. In *2011 IEEE International Conference on Robotics and Automation (ICRA'11)*, pages 1875–1882. IEEE.

Vidal-Calleja, T., Su, D., De Bruijn, F., and Miro, J. V. (2013). Learning spatial correlations for bayesian fusion in pipe thickness mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 683–690.

Walter, M. R., Eustice, R. M., and Leonard, J. J. (2007). Exactly sparse extended information filters for feature-based slam. *The International Journal of Robotics Research*, 26(4):335–359.

Wang, J. and Englot, B. (2016). Fast, accurate gaussian process occupancy maps via test-data octrees and nested bayesian fusion. In *ICRA*, pages 1003–1010.

Wellington, C., Courville, A. C., and Stentz, A. (2005). Interacting markov random fields for simultaneous terrain modeling and obstacle detection. In *Robotics: Science and Systems*, volume 6, pages 1–8.

Whittle, P. (1963). Stochastic-processes in several dimensions. *Bulletin of the International Statistical Institute*, 40(2):974–994.

Wijerathna, B., Vidal-Calleja, T., Kodagoda, S., Zhang, Q., and Miro, J. V. (2013). Multiple defect interpretation based on gaussian processes for mfl technology. In *SPIE Smart Structures and Materials+ Nondestructive Evaluation and Health Monitoring*, pages 86941Z–86941Z–12. International Society for Optics and Photonics.

Wikle, C. K., Berliner, L. M., and Cressie, N. (1998). Hierarchical bayesian space-time models. *Environmental and Ecological Statistics*, 5(2):117–154.

Wilkinson, J. (1968). A priori error analysis of algebraic processes,. In *International Congress of Mathematicians*, page pp. 629âĂŞ640.

Williams, C. K. and Barber, D. (1998). Bayesian classification with gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–1351.

Williams, C. K. and Rasmussen, C. E. (1996). Gaussian Processes for regression. *Advances in Neural Information Processing Systems*, pages 514–520.

Williams, C. K. and Seeger, M. (2001). Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems*, pages 682–688.

Williams, O. and Fitzgibbon, A. (2007). Gaussian process implicit surfaces. *Gaussian Proc. in Practice*, pages 1–4.

Williams, S. B., Dissanayake, G., and Durrant-Whyte, H. (2002). An efficient approach to the simultaneous localisation and mapping problem. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 1, pages 406–411. IEEE.

Woodbury, M. A. (1949). The stability of out-input matrices. *Chicago, IL*, 9.

Wurm, K. M., Hornung, A., Bennewitz, M., Stachniss, C., and Burgard, W. (2010). Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems. In *Proc. of the*

*ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*, volume 2.

Ye, C. and Borenstein, J. (2003). A new terrain mapping method for mobile robots obstacle negotiation. In *In Proc. of the UGV Technology Conference at the 2002 SPIE AeroSense Symposium*, pages 21–25.