

Non-parametric Workspace Modelling for Mobile Robots using Push Broom Lasers

Mike Smith
Wadham College



Supervisor:
Dr Paul Newman

Robotics Research Group
Department of Engineering Science
University of Oxford

October 2011

Abstract

This thesis is about the intelligent compression of large 3D point cloud datasets. The non-parametric method that we describe simultaneously generates a continuous representation of the workspace surfaces from discrete laser samples and decimates the dataset, retaining only locally salient samples. Our framework attains decimation factors in excess of two orders of magnitude without significant degradation in fidelity.

The work presented here has a specific focus on gathering and processing laser measurements taken from a moving platform in outdoor workspaces. We introduce a somewhat unusual parameterisation of the problem and look to Gaussian Processes as the fundamental machinery in our processing pipeline. Our system compresses laser data in a fashion that is naturally sympathetic to the underlying structure and complexity of the workspace. In geometrically complex areas, compression is lower than that in geometrically bland areas. We focus on this property in detail and it leads us well beyond a simple application of non-parametric techniques. Indeed, towards the end of the thesis we develop a non-stationary GP framework whereby our regression model adapts to the local workspace complexity.

Throughout we construct our algorithms so that they may be efficiently implemented. In addition, we present a detailed analysis of the proposed system and investigate model parameters, metric errors and data compression rates. Finally, we note that this work is predicated on a substantial amount of robotics engineering which has allowed us to produce a high quality, peer reviewed, dataset - the first of its kind.

Statement of Authorship

This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

Mike Smith, Wadham College

Funding

This work was funded through an Engineering and Physical Sciences Research Council CASE award with BAE Systems, and by the European Commission under grant agreement number FP7-231888-EUROPA. UK EPSRC (CNA and Platform Grant EP/D037077/1).

Acknowledgements

Oxford is fantastic. The people, the city and certainly the institution have taught me so much, but there are some people without whom I would not have had this experience. Foremost my supervisor, Paul who has had such faith, patience and support for me throughout some exciting and trying times. I feel he has taught me far more than just robotics – not least, improving my literacy skills.

Thank you also to the Mobile Robotics Group for some brilliant discussions, often late into the night, and to David Edwards who it is a privilege to know.

To all my friends that have “hung on in there” while I have indulged in this adventure and to my family for being the forever loving and caring pillars of my life. It means so much to me. Finally to a love that has escaped me but who has taught me so much about life along the way.

Contents

1	Introduction	1
1.1	Contributions	4
1.2	Literature Review	6
1.2.1	Representation of Laser Data	6
1.2.2	Reducing Redundancy and Inconsistency in Laser Data	10
1.2.3	Surface Modelling with Gaussian Processes	12
1.3	Thesis Structure	14
1.4	Publications	16
2	Robotic System and Data Generation	17
2.1	Sensor Data Description	18
2.1.1	Coordinate Frames	20
2.2	Workspace Summary	23
2.2.1	Partitioning and Event Tagging	24
2.3	Data Access Methods	29
2.3.1	Supplied Tools	29

2.4	Data Visualisation	30
2.4.1	Combining Laser and Wheel Odometry	32
2.4.2	Laser, Wheel Odometry and Back Projected Colour	34
2.4.3	Visual Odometry	37
2.4.4	Trajectory Relaxation using Visual Odometry, Place Recognition and Laser Data	38
2.4.5	Laser Data using Relaxed Trajectory Estimates	41
3	Representations Of Laser Data For Push Broom Laser Systems	44
3.1	New Nomenclature Introduced in this Chapter	46
3.2	Global Representation Using Pose Estimates	47
3.2.1	Assessing Global Map Consistency	48
3.2.2	Discussion	56
3.3	Sensor Configuration Space parameterisation	58
3.4	Laser Data in Sensor Configuration Space	62
3.5	Assessing Algorithm Accuracy in Sensor Configuration Space	68
3.6	Summary	72
4	Bayesian Regression Techniques	74
4.1	New Nomenclature Introduced in this Chapter	76
4.2	Bayes Rule	77
4.2.1	Gaussian Variables with Linearly Dependent Means	78
4.3	Bayesian Parametric Regression	80
4.4	Gaussian Process Regression (GPR)	84

4.4.1	Hyperparameters	89
4.4.1.1	Mean Function	90
4.4.1.2	Covariance Function	90
4.4.2	Practical Considerations for Covariance Matrices	94
5	Gaussian Process Regression in Sensor Configuration Space	97
5.1	Model Assumptions for GPs in Sensor Configuration Space	98
5.2	Simple Stationary Covariance Function for Push Broom Laser Systems	99
5.3	Is this the Right Covariance Function?	101
6	Surface Mapping And Decimation In An Active Window Framework	103
6.1	New Nomenclature Introduced in this Chapter	104
6.2	Algorithm Overview	105
6.3	Active Windowing	107
6.3.1	Online Predictions	110
6.3.2	Offline Predictions	110
6.4	Active Sampling through Kullback-Leibler Divergence	111
6.5	Parametric Mean Function	115
6.6	Implementation Considerations	118
6.6.1	Updating Cholesky Factors	120
6.6.2	Updating Forward Substitutions	123
6.6.3	Mean Function Conditioning	125
6.6.4	Forming Predictions during Introspection	126
6.7	Experimental Setup	126

6.8	Results	129
6.8.1	Compression Rate	129
6.8.2	Surface Error	132
6.8.3	Timing Considerations	135
6.8.4	Choosing Hyperparameter Values	137
6.8.5	Varying the Regression Model	137
6.9	Conclusions	141
7	Non-Stationarity in an Active Window Framework	143
7.1	New Nomenclature Introduced in this Chapter	145
7.2	Non-Stationarity through a Data Dependent Gaussian Mixture Model .	146
7.3	Bayesian Approach to Non-Stationarity	147
7.3.1	Log Marginal Likelihood of Parametric Mean GPs	150
7.3.2	Simple Bayesian Quadrature	151
7.3.3	Bayesian Quadrature for Gaussian Process Regression	156
7.3.4	Modifications to the Covariance Function	158
7.4	Simple Approximations to Predictive Bayesian Quadrature	160
7.4.1	Choosing Hyperparameter Samples	163
7.4.2	Weighting Via Re-Normalised Marginal Likelihood	165
7.4.3	Discretised Type II Maximum Likelihood (ML-II) Estimate . . .	166
7.5	Non-Stationary Results	167
7.5.1	Comparing Approximate GMMs	167
7.5.2	Comparing Surface Error	172
7.6	Discussion	172

7.7	Conclusions	176
8	Conclusions	178
8.1	Future Work	180
8.1.1	Implementation Speed-ups	181
8.1.2	Alternative Models	183
8.1.3	Alternative Laser Sensor Configuration Spaces	185
8.1.4	Combining Sensor Configuration Spaces	187
8.1.5	Useful Framework By-Products	189
8.2	Concluding Remark	190
A	Research Platform Details	191
A.0.1	Finishing the Robotics Research Platform	192
A.0.2	Software Considerations	194
B	Pose Relaxation	198
C	3D Scan Matching via Iterative Closest Point	203
C.1	ICP stages	205
C.2	Comparison of Approaches	210
D	Properties of Gaussian distributions	212
D.1	Joint Gaussian distributed variables	213

Nomenclature

Laser sensor model

$\mathcal{W}$ - 3D Euclidean workspace

$\mathbf{x}$ - a single point in 3D Euclidean workspace $\mathbf{x} \in \mathcal{W}$

$\mathcal{Q}$ - sensor configuration space

θ - angular position of laser beam relative to the laser sensor

t - timestamp of laser range measurement

$\mathbf{q}$ - sensor configuration $\mathbf{q} = [\theta, t]^T$ in sensor configuration space $\mathbf{q} \in \mathcal{Q}$

ψ_θ - angular scale factor so that laser measurements form a unit spaced grid in $\mathcal{Q}$ ($\psi_\theta = 0.5^\circ$ – the angular resolution of the scanner)

ψ_t - times scale factor so that laser measurements form a unit spaced grid in $\mathcal{Q}$ ($\psi_t = \frac{1}{75}$ seconds – the temporal resolution of the scanner)

$\mathbf{p}$ - 6 DOF pose of the sensor's laser beam

N - number of laser measurements in a laser data dataset

$\mathbf{r}$ - all laser range measurements in a dataset $\mathbf{r} = [r_1, r_2 \dots r_N]^T$

$\mathcal{Q}$ - set of all sensor configurations in a dataset (training data) $\mathcal{Q} = \{\mathbf{q}_i\}_{i=1}^N$

$\mathcal{D}$ - set of all laser measurements $\mathcal{D} = \{(\mathbf{q}_i, r_i)\}_{i=1}^N$

$G(.)$ - mapping that maps a sensor configuration $\mathbf{q}$ to a range measurement r

$V(.)$ - mapping that maps a sensor configuration $\mathbf{q}$ to a vehicle pose $\mathbf{p}$

$E(.)$ - mapping that maps a vehicle pose $\mathbf{p}$ and a laser range measurement r to a 3D point $\mathbf{x}$ in Euclidean workspace $\mathcal{W}$

Gaussian Process Regression (GPR)

- $\cdot^*$ - subset of a GP concerning input space $\mathcal{Q}$ that are used for predictions
- $\cdot_0$ - vector of infinite length (a function evaluated at all possible locations in its input space)
- $\mathcal{GP}(\mu(\cdot), k(\cdot, \cdot'))$ - Gaussian Process (GP) fully specified by mean function $\mu(\cdot)$ (expectation) and covariance function $k(\cdot, \cdot')$ (variance)
- ξ - regression model hyperparameters
- ζ - regression model parameters - the underlying function $\zeta(\mathbf{q})$
- $p(\zeta|Q)$ - Gaussian Process (GP) prior of the underlying function $\zeta \triangleq \zeta(Q)$ evaluated at the set of training sensor configurations Q
- $p(\mathbf{r}|\zeta, Q)$ - likelihood of range measurements $\mathbf{r}$ given an additive iid Gaussian regression model and underlying function $\zeta(\mathbf{q})$ observed at Q .
- $p(\mathbf{r}|Q)$ - marginal likelihood range measurements $\mathbf{r}$ given an additive iid Gaussian regression model and marginalised underlying function
- $p(r^*|\mathbf{q}^*, \mathcal{D})$ - posterior predictive distribution of range measurement r^* given sensor configuration $\mathbf{q}^*$, data $\mathcal{D}$ and an additive iid Gaussian regression model
- $K(Q, Q)$ - covariance matrix between training data Q (positive semi-definite)
- $K_{\mathbf{r}}$ - augmented covariance matrix $K_{\mathbf{r}} = K(Q, Q) + \sigma_m^2 I$ (positive semi-definite)
- $\mathbf{k}(Q, \mathbf{q}^*)$ - covariance between training points Q and test point (prediction) $\mathbf{q}^*$
- $k(\mathbf{q}^*, \mathbf{q}^*)$ - covariance (variance) of test point (prediction) $\mathbf{q}^*$
- σ_m^2 - measurement variance (sensor accuracy in the regression model)
- σ_p^2 - process noise (scales output space of the covariance function $k(\cdot, \cdot')$)
- l - length scale (scales input space of the covariance function $k(\cdot, \cdot')$)

Active Sampling

- $\tilde{\cdot}$ - elements within the current active set
- n - number of measurements in the active set of measurements
- η - span of the active window (index of the newest measurement to have been considered, minus the index of the oldest measurement in the active set of measurements $\tilde{\mathcal{D}}$)
- α - minimum number of measurements between the oldest measurement in $\tilde{\mathcal{D}}$ and online predictions to ensure adequate conditioning
- $\mathcal{D}'$ - set of all laser measurements that have been actively selected
- $D_{KL}(\cdot||\cdot)$ - Kullback-Leibler (KL) divergence
- κ - User selected KL threshold. Sets the trade off between the accuracy of our representations versus the number of retained laser measurements

Parametric Mean GP

- $Z(\cdot)$ - zero mean GP regressor $Z(\cdot) \sim \mathcal{GP}(0, k(\cdot, \cdot))$
- $\mathbf{h}(\cdot)$ - basis functions $[1, \theta, \theta^2, \theta^3, t, t^2, \theta t, \theta t^2, \theta^2 t, \theta^2 t^2]^T$
- e - number of basis functions
- $\boldsymbol{\beta}$ - basis function weightings $\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\beta}; \bar{\mathbf{b}}, B)$
- $S(\cdot)$ - parametric mean GP regressor $S(\cdot) = Z(\cdot) + \mathbf{h}(\cdot)^T \boldsymbol{\beta}$
- H - stacked basis functions $[\mathbf{h}(\mathbf{q}_1), \mathbf{h}(\mathbf{q}_2) \dots \mathbf{h}(\mathbf{q}_n)]$
- $\bar{\boldsymbol{\beta}}$ - mean of posterior distribution of basis function weightings $\boldsymbol{\beta}$ with respect to ranges. For a uniform $\boldsymbol{\beta}$ ($B^{-1} \rightarrow \mathbf{0}$) $\bar{\boldsymbol{\beta}} = A^{-1} H K_{\mathbf{r}}^{-1} \mathbf{r}$
- A - covariance of posterior distribution of basis function weightings with respect to ranges $A = H K_{\mathbf{r}}^{-1} H^T$ (positive semi-definite)
- $\mathbf{u}$ - premultiplier for $\boldsymbol{\beta}$ in parametric GP predictions
- $\mathbf{u} = \mathbf{h}(\mathbf{q}^*) - H K_{\mathbf{r}}^{-1} \mathbf{k}(Q, \mathbf{q}^*)$

Non-stationary Gaussian Process Regression

- $\cdot^*$ - subset of a GP concerning input space $\mathcal{E}$
- g - number of hyperparameter samples ξ_i
- $\mathcal{E}$ - hyperparameter sample space
- ξ_i - hyperparameter sample $\xi_i \in \mathcal{E}$
- Ξ - set of all hyperparameter samples $\Xi = \{\xi_i\}_{i=1}^g$
- $\phi(\xi)$ - GP Prior over marginal likelihood $p(\mathbf{r}|Q, \xi)$
- $v(\xi)$ - GP Prior over posterior predictive distribution $p(r^*|\mathbf{q}^*, \mathcal{D}, \xi)$
- $p(\xi)$ - prior over hyperparameter space $\mathcal{E}$
- $p(r^*|\mathbf{q}^*, \mathcal{D})$ - marginalised posterior predictive distribution
- $p(r^*|\mathbf{q}^*, \mathcal{D}, \Xi)$ - approximation of marginalised posterior distribution as given
by Bayesian Quadrature approach
- μ_{GMM} - mean of our Gaussian Mixture Model (GMM) approximation
- σ_{GMM}^2 - variance of our Gaussian Mixture Model (GMM) approximation
- ρ - mixing coefficients (posterior predictive distribution weightings)
- ξ_{ML-II} - single hyperparameter sample with maximum marginal likelihood
 $(\max_i p(\mathbf{r}|Q, \xi_i))$ as used in our simple approximation of ρ
- $p(\xi_i|\mathcal{D})$ - posterior of the marginal likelihood under a uniform prior
 $(\rho_i = p(\xi_i|\mathcal{D}))$ as used in our approximation of ρ

Chapter 1

Introduction

Over the last decade 3D laser acquisition systems have permitted the visualisation of increasingly large and complex outdoor scenes. Indeed, they have become integral to many applications across disciplines including: surveying, urban planning, robotics, manufacturing, autonomous vehicles, architecture, gaming and entertainment, heritage preservation, and highway maintenance. Overleaf is an example of a typical laser point cloud that have become ubiquitous in these fields.

Whilst visually appealing when rendered, laser point clouds have limitations. First, each point represents a single discrete sample of the workspace as observed by the laser acquisition system. Although, as humans, we are able to recognise the workspace surfaces embodied by these laser measurements, point clouds do not readily admit access to the continuous nature of these underlying surfaces. Further processing is required to reason about locations on these surfaces anywhere other than where the measurements were taken. Second, point clouds are inefficient because they contain redundant



Figure 1.1: Laser point cloud of the New College quadrangle, Oxford, using data collected from our robotics research platform. As humans we can easily parse the workspace surfaces that are present in the laser data, but can a more efficient representation be created that can permit direct and automatic access to these underlying surfaces?

samples that, given neighbouring samples, convey little additional information about the underlying scene geometry. 20 million laser measurements were used in the above image; many of these encode the central lawn area of the quadrangle that could, for example, be encoded by a single opaque ellipse.

This thesis describes a non-parametric method which generates a continuous representation of the workspace from discrete laser samples and simultaneously decimates the dataset, retaining only locally salient samples. These retained samples are all that are required in order to represent our continuous surface. The specific focus of this

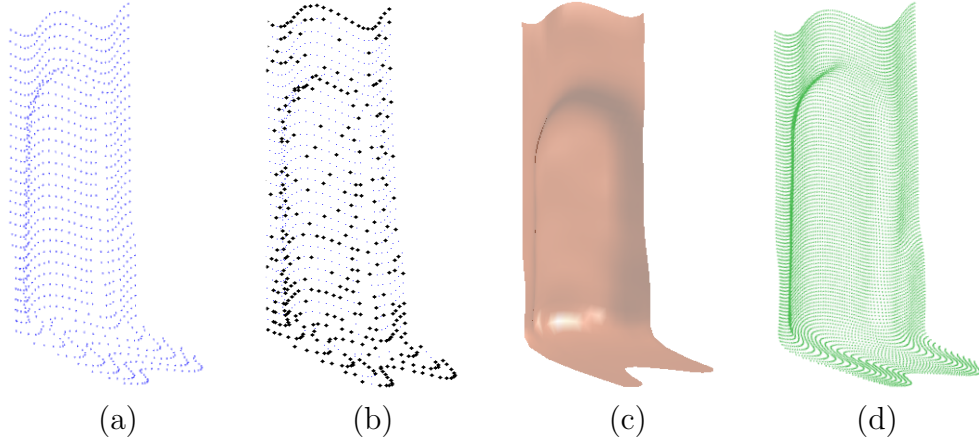


Figure 1.2: Main contributions of this thesis. (top) An image taken from a camera mounted atop our mobile robotics platform showing the central archway of Figure 1.1 from the side. (bottom) Visualisations of the closed door one third along the wall from the left of the image. (a) Raw laser data that is acquired from our laser system on the side of our robotics platform. (b) Data that is automatically selected in the construction of our underlying workspace surfaces are plotted in black. (c) Mean prediction of our workspace surface using only the subsampled data. (d) Supersampling of the original point cloud using our algorithm’s ability to form predictions at arbitrary locations on the estimated workspace surface.

work lies in clouds that are generated from mobile robotics platforms as they explore the workspace that they are situated in. We have found that we may decimate point clouds in excess of two orders of magnitude without significant degradation in fidelity. An outline of our approach is provided in the above figure.

At the heart of our method is the application of a Gaussian Process (GP) to form predictions at arbitrary locations along the underlying workspace surfaces that are conditioned on the laser measurements. We shall discuss GPs in detail in Chapter 4, but for now all that needs to be known is that they are a powerful and flexible means to perform our surface regression in a probabilistic framework. They are fully specified by a mean and covariance function and result in a continuous, implicit and non-parametric representation of the underlying workspace surface.

We decimate point clouds through the use of a GP to generate a predictive distribution for each of the laser samples. By evaluating the similarity between the distribution of a measurement (using a sensor model) and its corresponding GP-predicted distribution, we arrive at a criterion for keeping or dropping these samples. We refer to this process as *Active Sampling* – the GP framework actively guides us in deciding which of many laser measurements should be retained and used later for surface reconstructions.

1.1 Contributions

This thesis makes the following contributions:

Novel laser parameterisation that permits the representation of vertical and overhanging structures such as walls and trees. To the best of our knowledge this is not possible with current GP terrain mapping algorithms. The parameterisation may be trivially extended to work with data generated beyond that of the simple push broom laser system considered in this thesis. This parameterisation also decouples vehicle trajectory estimates from that of our mapping algorithm thereby allowing the two to be

computed independently.

Active subsampling of laser data using an information theoretic criterion that incorporates prediction uncertainty in addition to predicted values. This can be controlled through a single divergence threshold parameter (with units in *nats*) that sets the resulting trade off between the accuracy of the representation versus the number of retained laser measurements. Its value is entirely at the user’s discretion.

Online and offline surface predictions may be produced by our algorithm; the former when actively sampling the workspace as laser data is collected, the latter when using just the retained measurements. These surface predictions permit us to make predictions anywhere on our underlying surface, rather than being restricted to the laser data sample sites.

Non-parametric regression model through an efficient implementation of a parametric mean Gaussian Process that uses an *active window* to ensure that computational requirements are bounded. Here predictions are made using support from a finite, time varying region of the workspace, and we reduce the algorithm complexity of our approach by reusing computations whenever possible.

Self selection of model parameter values through a general GP framework. This framework permits us to use any stationary covariance function and produces an algorithm that is overall non-stationary. Broadly speaking this means that we allow our regression model to vary across the workspace, and we do so by using a measure of fit

(the Marginal Likelihood) of the local laser measurements given our model.

World’s first Robotics data paper that resulted from the creation of a dynamically balanced robotics platform. This has stimulated further releases by the robotics community to promote the much needed sharing of robotics data.

1.2 Literature Review

We first overview various representations of laser data (Section 1.2.1). In Section 1.2.2 we then discuss techniques that reduce two problematic properties that are commonly found in robotics laser data, namely redundancy and inconsistency. We then consider surface representations that use GPs in more detail in Section 1.2.3.

1.2.1 Representation of Laser Data

Within the robotics community, 3D surface reconstruction from sparsely sampled environments is commonly motivated by the need to provide an accurate reconstruction of the workspace across scales. This is commonly achieved by meshing techniques (see, for example, [30, 43, 90]) whereby every datum of a point cloud is represented by the vertex of a polygon, typically triangles. Often such techniques are augmented by a general mesh decimation step which aims to obtain a good representation of the environment (through optimisation with respect to approximation error), while at the same time reducing the amount of data to be stored (see [39] or [18] for an overview). Commonly, mesh creation, as well as the reduction in the number of vertices, is also based on sensor and environment dependent heuristics. Problems are encountered when the data arise

from a non-homogeneous sampling of the workspace and, or coverage is incomplete – both are frequently the case in our problem domain.

Efficient surface reconstruction has been extensively studied in the last decade in the Graphics and Computer Vision communities ([22] provides a high level overview). Interpolation methods for unordered [5, 32, 6] and ordered [8, 11] laser points typically suffer from undesirable artifacts [3]. They are likely to be rough, containing bumps and other kinds of undesirable features, such as holes and tunnels which require an additional smoothing step such as [26, 104, 63]. These Voronoi inspired techniques are also fragile to non-uniform sampling due to their reliance on a number of heuristics to guide their construction and tend to be computationally expensive. Alternative solutions globally smooth the meshes while they are being constructed either by averaging points to make mesh vertices [102], or by averaging overlapping surfaces [108]. These approaches are not appropriate for online mobile robotics applications as they require the replacement of highly accurate measurements of the workspace with aesthetically pleasing, but sometimes misleading data points. There have also been approaches that use global implicit surfaces, for example [47, 16]. The latter is based around Radial Basis Functions (RBF) which tend to be computationally demanding. Since these algorithms are global in their nature they require input from all laser points. They are therefore not applicable to our surface reconstruction problem where we do not know all laser measurements a priori.

Efficient reconstruction from high density data has also received much attention [75, 62, 58]. Local implicit reconstructions are commonly based on localised RBFs (see [75] for an overview) or a technique called Moving Least Squares (MLS) as both tend

to be more robust to noise than other methods. MLS was first introduced by [62] for drawing contours whereby simple local regression curves were fitted to data. The weighting on each sample point decreases with distance from the centre of the local area of interest (see [54] for an overview and [34] for a novel extension that uses spheres). Although these techniques have some “hole filling” capabilities they concern large, high density point clouds. Efficiency is therefore often of prime importance and not surface accuracy or the handling of incomplete or sparse data. We require our algorithm to be robust to all these issues for the data that is considered in this thesis.

Alternative approaches consider parametric representations based on geometric primitives [36, 106, 38]. Planar representations are a popular choice since planes are pervasive in man-made environments. However, the reliance on geometric primitives places strong assumptions on the workspaces which are rarely justified in complex outdoor environments. Non-Uniform Rational B-Splines (NURBS) provide a powerful smooth surface representation of arbitrary topology (see [82] or [83] for a detailed explanation). However they require careful partitioning of the data into NURB patches which must be stitched together using geometric continuity conditions. In general these operations require complex high-dimensional constrained optimisation. Our approach permits a similar if not greater degree of flexibility through the use of GPs, without the need for optimisation.

A number of approaches construct their representation in relation to a small, base, subset of data [41, 8, 50]. Smooth parameterisations of varying complexity can then be constructed using these coarse base meshes. In some algorithms the fixed spacing in the base mesh can limit the extent that small features in the data can be accurately

resolved without reconstructing the base mesh with additional triangles [41, 8]. Although [50] addresses this weakness there can also be problems associated with surface representations locking into the wrong surface in regions of high curvature (due to the smoothness of the initial mesh). Multiresolution Adaptive Parameterization of Surfaces (MAPS) [1] solves this problem through proceeding instead from a fine to a coarse base mesh. This approach also permits a hierarchical representation of the data, which like our approach (through our divergence threshold) allows the user to select the level of surface complexity. Although there has been careful consideration to ensure that both storage and computational requirements are minimised in the creation of the hierarchy, this capability provides a large overhead. The algorithm also does not address the problem of data storage reduction which is a requirement for our algorithm.

The requirement of robots to operate in ever larger environments provides ample incentive for research into suitable models of the terrain encountered. Consequently, much recent robotics research into 3D reconstruction techniques trace their origin to this domain. Common approaches include discrete 2D, 2.5D and 3D methods such as [59, 9, 80, 107, 115], which represent the world as a regular grid of cells or voxels. Each cell stores the probability of it being occupied. As new sensor measurements arrive these probabilities are continuously updated. These representations are dependent on grid resolution and suffer from the necessity to strike a trade-off between resolution (i.e. memory requirements) and volume covered. This is particularly problematic for the surfaces that we are concerned with generating since they are not bound in the volume that they occupy.

1.2.2 Reducing Redundancy and Inconsistency in Laser Data

There has been much work into “thinning” (or sparsifying) point clouds directly from the laser data ([4] provides a comprehensive overview). Here various approaches are used to determine which points within the cloud are redundant and can therefore be removed. For example, [19] propagate fronts from random sample points that travel at varying speeds according to a user-defined heuristic. The intersection of these fronts determine the location of important measurements. There has also been much work from the Graphics and Computer Vision communities that use MLS [79, 3, 56, 53]. The work of [79] lifts the restriction of purely sub-sampling from the point clouds, therefore avoiding aliasing artefacts and uneven sampling distributions through the repositioning of points. However, these thinning and point repositioning algorithms are driven by considerations that produce simple meshes rather than the preserving of the most salient measurements. In contrast, our approach selects data on the basis of their information content relative to surrounding measurements and does not resort to the creation of artificial measurements.

The point decimation considerations in [55, 46, 81] are driven from rendering considerations. They suggest that point data should be the universal meta-primitive due to their low relative storage overhead and ease of hardware acceleration. Hardware acceleration provides further advantages such as quick reasoning about occlusions (given a sufficient sampling density). Surfels [81] have recently been used in robotics applications [40] due to their compatibility with a new form of sensor, an RGB-D camera. This sensor is capable of simultaneously providing both range and colour information of points. Whilst these algorithms help to alleviate storage requirements and could

provide a quick means to view laser data, they do not permit access to the underlying workspace surfaces that the data captures.

All point clouds generated from real data suffer from inconsistencies that result from the imperfect alignment of laser scans. Laser data is collected in scans that are necessarily separated in time (with specific format dictated by the laser acquisition system). Without perfect information about the trajectory of the laser system between each scan, the rigid transform between scans must be found either through an estimate of the trajectory or an alignment technique. Alignment is commonly performed through techniques such as Iterative Closest Point (ICP) [12, 44, 17, 15] (see Appendix C for a detailed comparison of approaches). There have also been a number of techniques that aim to fuse (consolidate) point cloud inconsistencies into a single representation [42, 24, 113, 112, 28]. The first three use an implicit representation of the underlying surfaces based on a distance function defined on a volumetric grid (known as a *level set*). They extract an isocontour where the cumulative distance function is 0. Of these [24] can handle uncertainty in range data. Gaussian Processes have also been used effectively [112, 28] through assigning a covariance between measurements that have been taken from distinct sweeps of the workspace, and even across different sensor modalities. These algorithms perform well when the path of the laser scanner is known to a high level of accuracy and the workspace is static. However, for mobile robots the trajectory estimate is frequently the subject of errors, sometime grossly so (due to unmodelled motion like slip for example) as we shall see in Chapter 3. Inconsistencies in point clouds that are generated under these conditions are often too large to permit these fusing approaches to produce accurate surface reproductions.

Within the mobile robotics community there have been attempts to reduce point cloud discrepancies that result from poor vehicle trajectory estimates [76, 69, 33, 105, 48, 29]. These attempt to perturb, or “relax”, the estimates so that errors are spread along the kinematic chain of trajectory poses to produce a coherent metric map. We explore our work presented in [69] in Chapter 3. Based on our finding we produce a novel parameterisation that is independent of trajectory estimates, thereby side-stepping the problems associated with their estimation.

1.2.3 Surface Modelling with Gaussian Processes

In recent years there has been a trend to employ continuous, non-parametric, models of surfaces in the terrain modelling community. In this context, GP models are particularly favoured for their ability to handle incomplete data in a principled probabilistic fashion. Examples of such approaches include [52, 85, 35, 111]. They were first introduced to the field of Geostatistics under the name *Kriging* many years previously by [60] amongst others. While these approaches differ in their choice of (sometimes non-stationary) covariance function and GP sparsification method, they adopt the same parameterisation of the problem. They model a function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ which associates a single elevation value z with any given position (x, y) in 3D Euclidean space. This limitation is exploited in [35] by constraining surface regression with visibility information of the sensor. While this *well-formed* functional mapping is effective for terrain modelling it precludes application to our full 3D mapping case. We wish to model vertical and overhanging structures, such as walls and trees as we shall discuss in Chapter 3.

There has been much research into sparsification for GP regression. An overview is provided in [87]. Within this field, active subsampling strategies have been used to select information-rich data through use of information theoretic criteria [49, 25, 21, 95]. Our approach is particularly similar to [95]. However, we exploit the time-sequential nature of laser data to form an exact and inexpensive predictive distribution for use in our decision criterion.

Zero mean GPs are a popular choice in the robotics community [52, 85, 111, 71] and are justified through the desire to assume little prior knowledge and by symmetry arguments. A degree of smoothness is assumed (through an appropriate choice of smooth covariance function), whereby laser measurements spatially close to one another are assigned higher covariance than those further apart. As we shall discuss in Chapter 6, in the absence of data (which is frequently the case due to incomplete coverage of the workspace) however, smooth GP predictions tend towards their mean function (zero). They hence poorly predict the underlying workspace surface. Our framework concerns the decimation of point clouds and therefore this effect is exacerbated. A number of solutions have been suggested including the use of explicit basis functions to construct GP mean functions [70] and the somewhat anti-intuitive use of a non-smooth (and non-stationary) covariance function that is compactly supported [110]. The latter has been shown suited to data containing large jumps in the underlying workspace surface (that occur in for example terrains created by terrace mining). The smoothness of the resulting predictions are however heavily dependent on the extent of the envelope of the supporting measurements. This envelope varies dramatically with local sample density. Instead we adopt a parametric mean GP [14, 74]. Although this approach uses explicit

basis functions like [70], no additional parameters are required (as discussed in Chapter 6).

While we share the common goal of accurate 3D surface reconstruction with the literature discussed so far, our work bears closest relation to that of Gaussian Beam Processes [84]. The authors model laser data on a per-scan basis using a GP model to regress range on bearing. However, while our model also regresses on range, we achieve an implicit model of the entire workspace through a sliding window approach and active data selection. This provides significant advantages above and beyond a mere extension of Gaussian Beam Processes to the 3D case.

We also permit our GP regression model to vary automatically in response to local workspace complexity. Rather than relying on expert knowledge as in [85] we use a principled approach based on Bayesian Quadrature [73, 89, 77]. This results in a general framework that is not restricted in the type of model hyperparameters.

1.3 Thesis Structure

In Chapter 2 we detail the release of one of the world’s first robotics data papers. This release involved a substantial amount of systems engineering to create a robotics research vehicle capable of producing highly accurate and time synchronised data.

In Chapter 3 we analyse our dataset with the aim of understanding how to best represent underlying workspace surfaces that are present in the laser data. Specifically, we shall investigate the local and global consistency of laser point clouds and we shall

forge a simple and novel *local* parameterisation of laser data.

In Chapter 4 we provide the theory for Bayesian regression techniques that are used throughout this thesis. Gaussian Processes are discussed at length.

In Chapter 5 GPs are contextualised for our laser data parameterisation. Underlying assumptions are discussed as well as a GP covariance function that we have found pertinent to our regression.

In Chapter 6 we introduce our stationary GP in a windowed framework. In particular we detail our active sampling strategy, parametric mean function and implementation considerations that ensure an efficient algorithm. We then analyse our framework, paying attention to the surface reconstruction error of our predictions and the number of retained measurements.

In Chapter 7 our non-stationary extensions to our framework are discussed. Two alternative approaches are suggested and comparisons are made against our stationary GP.

In Chapter 8 we summarise our contributions and suggest a schedule for future work.

Throughout this thesis we have noted important new nomenclature at the beginning of each chapter and have adopted consistent notation that is commonly found in the GP community (see [88] for example).

1.4 Publications

This thesis has lead to the following publications:

Robotic Systems - [69, 67] - The International Journal of Robotics Research, November 2009 and In Proceedings of the International Symposium of Robotics Research, August 2009.

Data Paper - [97] - The International Journal of Robotics Research, May 2009.

Stationary GP Framework - [98] - In Proceedings of Robotics: Science and Systems VI, June 2010

Non-Stationary GP Extension - [100] - The International Journal of Robotics Research, June 2011.

Synopsis of GP Framework - [101] - In Proceedings of the International Joint Conference on Artificial Intelligence, July 2011. (best paper track).

Vision Guided Laser Segmentation - [99] - Towards Autonomous Robotic Systems, August 2010.

Chapter 2

Robotic System and Data Generation

The data paper mentioned in this chapter was published as ‘The New College Vision and Laser Dataset’ in the International Journal of Robotics Research, 2009.

In this chapter we present a large (30GB), free-to-access, dataset intended for use in mobile robotics research. We then provide an overview of how this dataset is used in this dissertation. The release followed a substantial technical endeavour that spanned the first 18 months of this DPhil. The result was a world class robotics research platform and a dataset that forms the basis for one of the first two *data papers*. We hope that data papers can collectively help to propel robotics research through the removal of the large time and financial barriers to entry that are associated with data capture.¹

¹As of writing there are 7 data papers. Our data paper has been cited 45 times since May 2009.

Gathered from a robot driving a couple of kilometers through a park and campus, the dataset contains 5 degree of freedom (DOF) dead-reckoned pose, laser range and reflectance data and 20Hz stereoscopic and omni-directional imagery. All data is carefully time stamped, all data logs are in human readable form and images are in standard formats. To facilitate the use of the dataset we also provide a set of data access tools and a synopsis of our data partitioning and tagging. Full documentation, including data samples, is available on the website <http://www.robots.ox.ac.uk/NewCollegeData/>.

We then illustrate how each of the sensor’s data streams can be combined, with a particular focus on producing global 3D laser maps. The techniques that are discussed form an overview of the background material which are used to visualise laser data in the following chapters. We also use them in the analysis of the quality of our robotic mapping system, to inform the choice of our laser data parameterisation in Chapter 3.

2.1 Sensor Data Description

Our anticipated consumers are parties interested in outdoor 6 DOF navigation and mapping (metric or topological) using vision and, or laser. Data collection was performed using the vehicle shown in Figure 2.1 while traversing through a college’s campus and adjoining park. The drive unit is that of a RMP200 base from Segway. It provides roll and pitch data from its internal IMU in addition to wheel encoder measurements which we combine into a 5 DOF odometry (x,y, yaw,roll and pitch). The vehicle’s sensor payload provides stereo image pairs gathered at 20Hz, 5-view omni-directional imagery and data from two lasers that scan at 75Hz and comprise of both range and



Figure 2.1: Data capture vehicle. A two wheeled dynamically balancing robot built around a RMP200 drive unit from Segway. Within the Dural mainframe there is an independent sensor payload consisting of battery packs, four networked PCs, a remote operation module and interface and power conditioning hardware for the vehicle’s sensor suite mounted on the vehicle’s periphery (see Appendix A for further details).

reflectance data. A breakdown of the complete sensor configuration, which we used for the entire 2.2km expedition (spanning 45 minutes), is provided in Table 2.1. All data was logged using the MOOS software infrastructure [68] and all data streams have synchronised timestamps to three decimal places of accuracy using a predecessor of the TicSync software [37]. Synchronised timestamps are crucial in allowing multiple sensor streams to be combined as we have done in Section 2.4 and in subsequent chapters. Further details, in particular the author’s contributions to the completion of the logging

Table 2.1: Description of payload sensors and modes of operation

Sensor	Description	Log Format	Measurements
Odometry	28Hz roll and pitch from Segway base. Yaw and displacement integrated from wheel counts	plain text	70×10^3
Laser	2, SICK LMS 291-S14 scanning over 90° with temporal resolution $\psi_t = \frac{1}{75}$ seconds and angular resolution $\psi_\theta = 0.5^\circ$ (hence 13,575 measurements per second) in body-vertical plane on sides of vehicle	plain text 13 bit range and 8 bit reflectivity (intensity)	70×10^6 laser points
Stereo	Point Grey bumblebee 20 Hz 512×384 grayscale	lossless .png	50×10^3 pairs
Panoramic Camera	Point Grey LadyBug 2, 5 images per frame 384×512 colour at 3 Hz	.jpg	16×10^3 images
GPS	CSI Seres 5 Hz intermittent reception due to urban setting	plain text	12×10^3

platform, can be found in Appendix A.

2.1.1 Coordinate Frames

Figure 2.2 shows the vehicle with sensor coordinate frames superimposed upon it. The numerical values of these transformations were measured by hand and are given in Table 2.2. Where possible we have refined these values by, for example, projecting laser data into a global frame or into the images recorded by the cameras and then visually

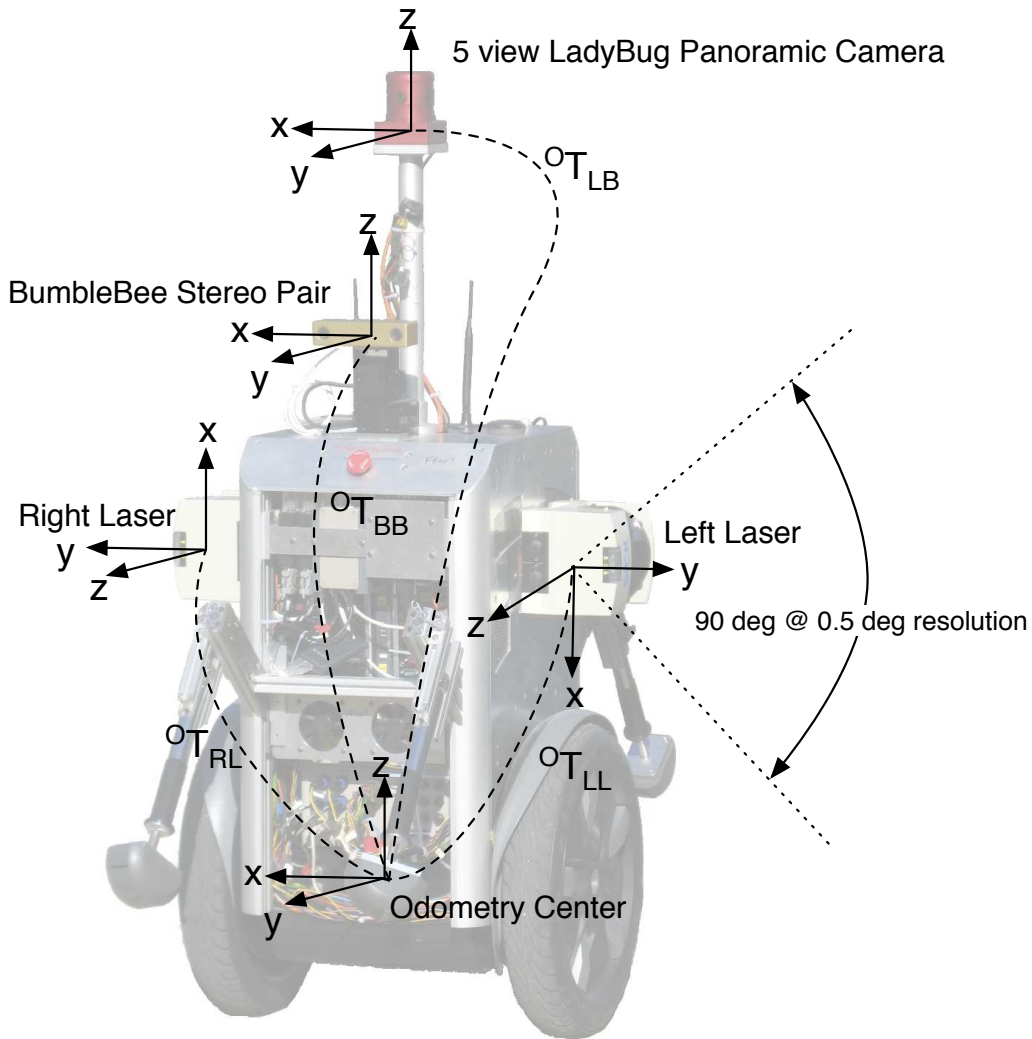


Figure 2.2: Vehicle with coordinate frames that depict sensor centres. Odometry data reports the robot centre position, taken to be a y -forward, z -up right-handed frame anchored half way between the platform's wheel axle. The Bumblebee's coordinate centre is on the left camera's focal point and the Ladybug's at the mean of the 5 radial camera centres. Laser scanner centres are at the point of intersection of the emitted beams.

checking discrepancies of features in the scene (see Section 2.4 for such projections). We have adopted a right handed, fixed axis “RPY” Euler angle parameterisation of $\mathcal{SO}_3$:

Roll Rotation by α about the body-y axis followed by

Pitch Rotation by β about the *original* body-x axis followed by

Yaw Rotation by γ about the *original* body-z axis:

$$R(\alpha) = \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} R(\beta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\beta) & -\sin(\beta) \\ 0 & \sin(\beta) & \cos(\beta) \end{bmatrix} R(\gamma) = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

such that the overall 3×3 RPY rotation matrix R is

$$R = R(\gamma)R(\beta)R(\alpha)$$

Table 2.2: Sensor Coordinate frames relative to the vehicle frame origin (taken as the odometry centre) with all angles $\{\alpha, \beta, \gamma\}$ in degrees and displacements $\{x, y, z\}$ in meters. For a visualisation of these transformations see Figure 2.2.

Transformation	Sensor	Estimate ($[x \ y \ z \ \alpha \ \beta \ \gamma]$)
${}^oT_{LB}$	LadyBug	$[0.000 \ 0.000 \ 1.169 \ 0 \ 0 \ 0]^T$
${}^oT_{BB}$	BumbleBee, left camera	$[-0.062 \ 0.205 \ 0.837 \ 0 \ -13 \ 0]^T$
${}^oT_{RL}$	Right Laser	$[0.270 \ -0.030 \ 0.495 \ -90 \ 0 \ -90]^T$
${}^oT_{LL}$	Left laser	$[-0.270 \ -0.030 \ 0.495 \ 90 \ 0 \ 90]^T$

2.2 Workspace Summary

The dataset was gathered within New College grounds during an afternoon early in November 2008. Figure 2.3 provides overhead imagery of the capture site, with a yellow line that depicts the extent of the vehicle’s exploration. The “Campus” environment is typical of buildings in Oxford being that of a grouping of Medieval buildings with similar architecture. The “quad” is an enclosure centred around an oval lawn. A short tunnel leads from the quad to a more open, cambered tarmac space surrounded by old buildings. This in turn leads, via a gate in large wrought iron screen (railings), to parkland (see Figure 2.9). The “Parkland” environment can be circumnavigated via a cambered gravel path around its perimeter. It is adjacent to an old city wall and a herbaceous border and occasionally passes under heavy autumnal foliage.



Figure 2.3: An aerial view of the workspace and paths driven (multiple loops are not shown). Although the data is contiguous, it can be partitioned into three epochs with different characteristics – the areas in which the vehicle was operating during each of these is marked in the figure. See Tables 2.3, 2.4 and 2.5 for a broad description of the epochs. Substantial details are available on the dataset website.

2.2.1 Partitioning and Event Tagging

We have broadly partitioned the data into the following three epochs – each with a particular character.

Epoch A “Campus”: three circumnavigations of the main quad followed by two more that also passed through a tunnel to the mid-section of the dataset. The epoch ends with a prolonged period near the park gates.

Epoch B “Parkland”: two circumnavigations of New College gardens. To the North West the path is predominantly flat with one large puddle. The South Eastern section of the path is partially covered with undergrowth, has some dark images and dips roughly 1m lower than the rest of the dataset.

Epoch C “Campus and Parkland” : one circumnavigation of both the “Campus” and “Parkland” environments. There are park gates at the beginning and end of the epoch. Sunlight was fading and so the images are darker than in earlier passes.

Figure 2.3 shows the epoch boundaries while Tables 2.3, 2.4 and 2.5 provide an overview of typical salient features of each of the epochs. The dataset website provides a substantially more detailed description of interesting and challenging events occurring in each epoch and provides time indexes by which they can be accessed. These data tags are listed both chronologically and by event type.

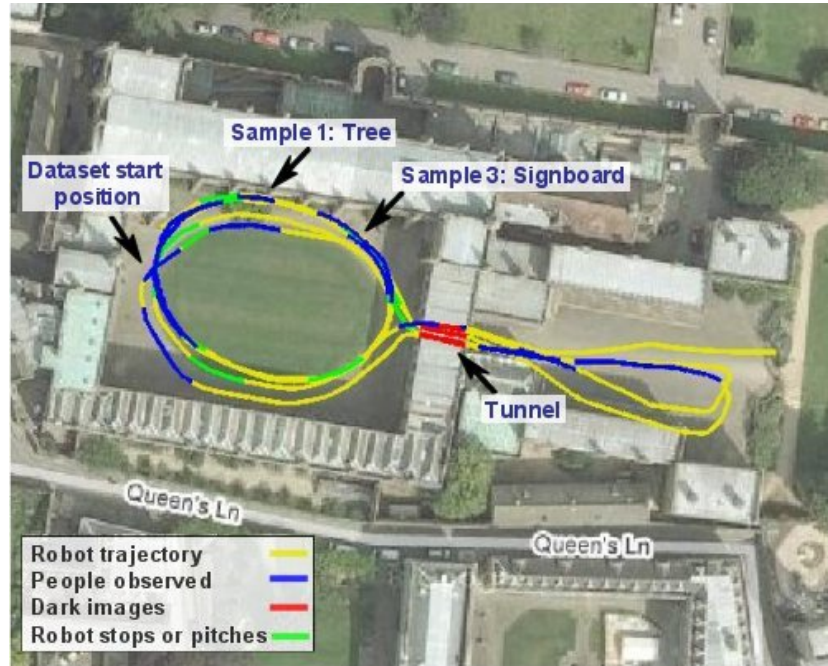


Table 2.3: Summary of events in Epoch A, the “Campus” environment - see Figure 2.3. Finer granularity is provided on the dataset website

Epoch A Start: 0.000 s Epoch A End: 1026.718 s	
Partition	Description
Quad Loop I Start:0.0 s End: 151.446 s	Platform remains stationary for 35 s before moving. Some pitching and rolling while turning. Groups of people are frequently observed walking towards the vehicle.
Quad Loop II Start:151.446 s End: 255.960 s	Tarmac ground has some puddles and wetted areas. Walking and standing groups of people are frequently observed.
Quad Loop III Start:255.960 s End: 425.161 s	Loose groups of people are observed. One group is observed both at the start and end of the loop.
Quad and Mid-Section I Start:425.161 s End: 655.580 s	Images are dark while the platform traverses a short tunnel. People often walk past the platform in the mid-section.
Quad and Mid-Section II Start:655.580 s End: 1026.718 s	Images in the tunnel section and under trees are dark. People are occasionally in view at a distance. Extended traversal through gates with some pitching while turning.



Table 2.4: Summary of events in Epoch B, the “Parkland” environment - see Figure 2.3. Finer granularity is provided on the dataset website

Epoch B Start: 1026.718 s Epoch B End: 1913.179 s	
Partition	Description
Park Loop I Start:1026.718 s End: 1518.990 s	Substantial platform pitching. Images are dark under foliage. A pair of people and a person pushing a cart overtake the platform.
Park Loop II Start:1518.990 s End: 1913.179 s	Platform pitches while navigating uneven terrain and slopes. People are rarely observed and are poorly illuminated.

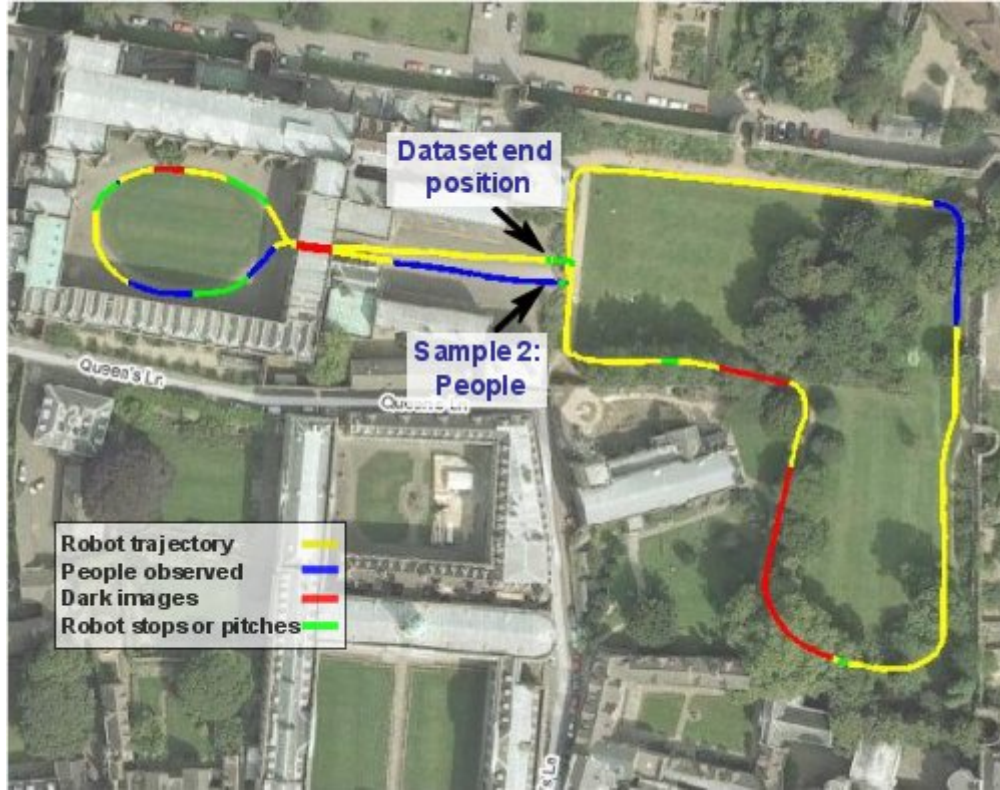


Table 2.5: Summary of events in Epoch C, the “Campus and Parkland” environment - see Figure 2.3. Finer granularity is provided on the dataset website

Epoch C Start: 1913.179 s Epoch C End: 2634.139 s	
Partition	Description
Quad and Mid-Section III Start:1913.179 s End: 2177.979 s	Prolonged navigation while entering the mid-section through gates. Images in the tunnel section are dark. The platform approaches a group of stationary people.
Park Loop III Start:2177.979 s End: 2634.139 s	The platform halts briefly during the loop and again at the end while turning out of the parks and through the park gates. Images under foliage are dark. A gardener is observed partially occluded by a bush.

2.3 Data Access Methods

The data is available free-of-charge from:

<http://www.robots.ox.ac.uk/NewCollegeData/>

This site contains many details of the dataset that are not appropriate for this summary. To avoid having to download the full 30GB just to evaluate the data, it is available in chunks covering several time scales from single short events (periods of a few seconds) up to the entire dataset. All non-image data is stored in a plain text files with a “alog” suffix. Each line contains a single log entry which is a comma separated list of token–value pairs. The format of all log entries is supplied in detail on the website.

2.3.1 Supplied Tools

Ease of use of the dataset is important for dissemination to the robotics community. While having plain text logs results in OS-independence and a degree of immediacy it has the disadvantage that a parsing step is required to extract numerical data. The distortion present in images are also unique to the camera used. Although there is nothing about our data that requires the use of our tools, we provide some to help with these tasks and to expedite data access. Specifically we have written dependency and OS free C++ source files which build tools to parse plain text logs into numerical arrays and to rectify the stereo images using the calibration parameters specific to our stereo pair. To demonstrate how the parser can be invoked from within MATLAB®[®], we also provide .m code that extracts laser and odometry streams, projects laser ranges into 3D and displays them in a MATLAB®[®] figure. The same code can also output laser points

and reflectances to plain text and VRML for use with third party viewers. Examples of results that can be obtained from our toolchain are presented in the following section and later throughout this dissertation.

2.4 Data Visualisation

We plot (x, y) coordinates for both GPS measurements and Segway odometry for the complete $2.2km$ dataset in Figures 2.4 and 2.5. GPS measurements provide a close representation of the true trajectory of the vehicle. When compared with the overhead imagery in Figure 2.3 one can clearly see the rough shape of the paths of the new college grounds. However, as commonly found in built up areas, GPS measurements suffer abnormal "ghost" points that are associated with multipath signals and a lack of line-of-sight to satellites (this is particularly evident in the tunnel in the centre of the "Campus" environment, roughly $(40, 0)$ in Figure 2.4).

Odometry results are also typical of dead-reckoned systems. Major sources of error include wheel slippage that compounds over time. Instruments on board the Segway base were also designed to allow the platform to dynamically balance and not for long term trajectory estimation (for example, the omission of an altimeter restricts trajectory estimates to only 5 DOF). The accuracy of the system could therefore have been bolstered through using a high precision IMU that would contribute less sensor noise in estimates. That said, this odometry is of significant quality to produce locally consistent maps (as we shall see in the following section) and we shall discuss local and global consistency more fully in Chapter 3.

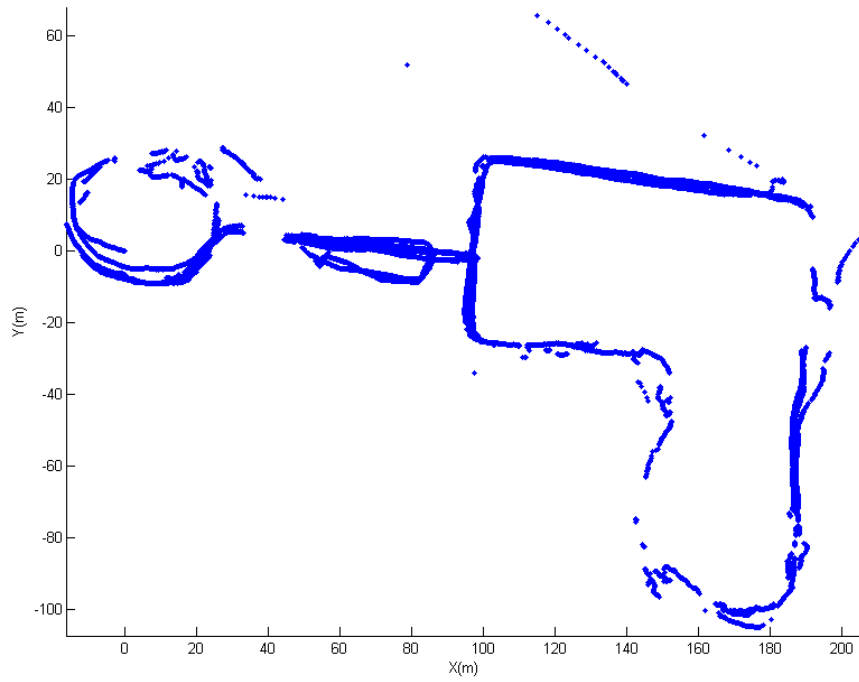


Figure 2.4: x, y location of all GPS fixes for the complete New College dataset taken on an overcast day in November. Coverage along the vehicle's trajectory is not complete due to poor visibility of satellites. There are also a number of anomalous results that are due to multipath signals.

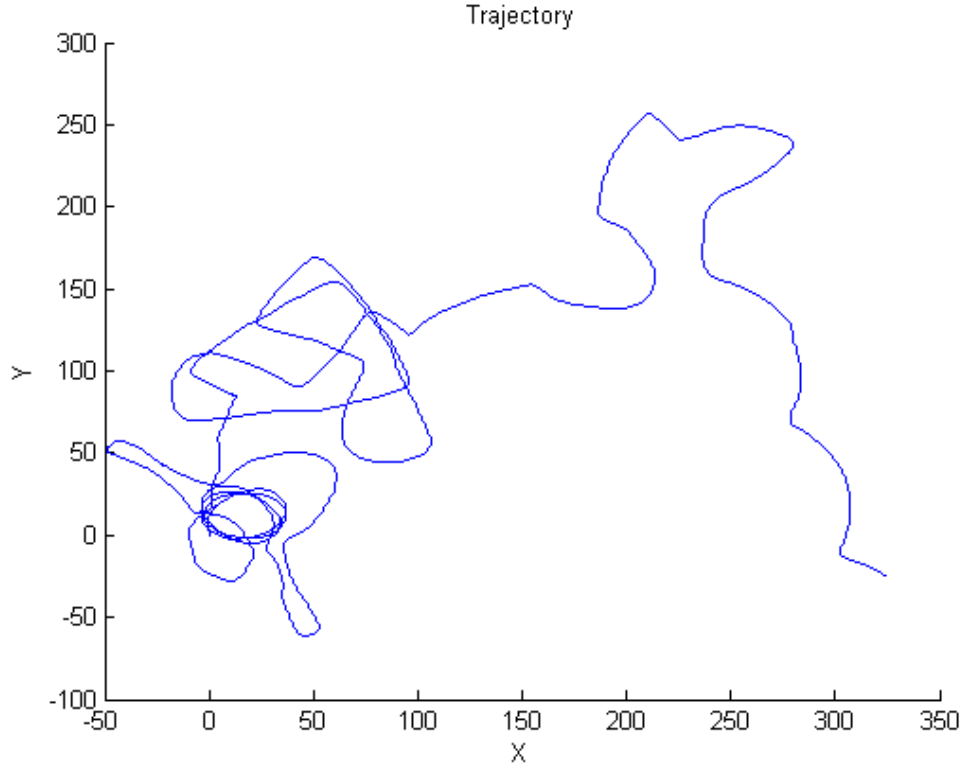


Figure 2.5: x, y location of all odometry poses for the complete New College dataset. Odometry was inferred through wheel encoders and the IMU on board the Segway RMP200 base. Over time considerable errors accumulate, primarily due to wheel slippage and the balancing nature of the research platform.

2.4.1 Combining Laser and Wheel Odometry

Of particular interest to this dissertation are the two SICK LMS 291-S14 laser scanners, mounted on the sides of the research platform and scan in a vertical plane perpendicular to the vehicles forward motion. This sensor configuration permits the simplest and cheapest of 3D data acquisition systems – the push broom laser (a general schema of such a system is shown in Figure 2.6). As each 2D laser scanner is pushed through the workspace the range measurements and the timestamp for each scan are recorded.

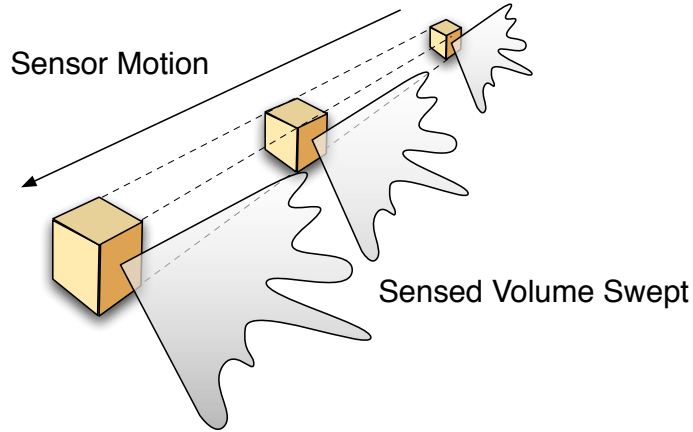


Figure 2.6: The geometry of a push broom laser. A 2D sensor is pushed through the environment, sweeping out a sensed volume.

Measurements are transformed into a local 2D sensor frame through a model of the laser sensor, and then into the local vehicle frame through ${}^O T_{RL}$ or ${}^O T_{LL}$ (Table 2.2) as appropriate. The timestamps are used to interpolate the odometry poses to produce a pose estimate at the time of the laser scan (assuming that the z coordinate is constant). Once transformed into the global frame through these pose estimates a 3D point cloud is then generated through aggregation of each scan. An example of the result of these operations is shown in Figure 2.7. On rendering we colour each laser measurement according to their reflectance which can provide additional scene understanding as shown by the clear lettering on the car number plate in Figure 2.8.

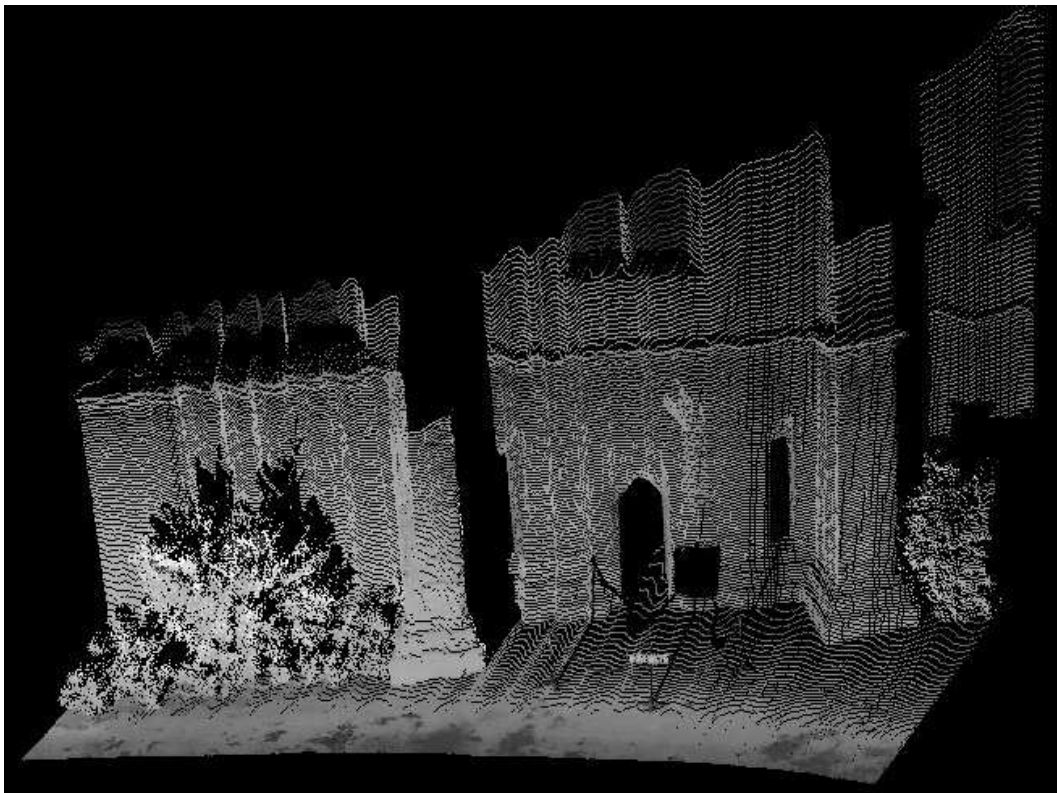


Figure 2.7: A detail in the New College quadrangle rendered relative to an odometry derived trajectory using laser range and reflectance data. This view was created by using our example code which extracts relevant data from a log file and writes the results to a VRML data file. This file was then loaded into a viewer and rendered. Note that by capturing reflectance data extra relief is added to the scene - foliage and the white writing on signs provide strong (bright) returns. The apparent wavy pattern on the walls is an illusion caused by the pitching of the vehicle as we shall discuss in Section 3.4.

2.4.2 Laser, Wheel Odometry and Back Projected Colour

We may also make use of colour information surrounding each laser measurement in the workspace. For this we back project laser measurements into images that are captured



Figure 2.8: Small section of laser data that embodies the front of a car. Data was generated using the tools detailed in Section 2.3.1. Colouring of points is supplied by laser reflectance. One can clearly discriminate coherent letters on the car number plate which are as a result from black lettering on a reflective backing.

by the omnidirectional camera mounted atop the research platform. We choose the image that has the closest timestamp to each measurement (see Figure 2.9 for a typical panoramic image). The transform between the two sensors are calculated through compositions such as

$${}^{RL_{t_1}}T_{LB_{t_2}} = {}^{O_{t_1}}T_{RL}^{-1} \cdot {}^{O_{t_1}}T_{O_{t_2}} \cdot {}^{O_{t_2}}T_{LB}$$

${}^{O_{t_1}}T_{O_{t_2}}$ represents the distance the research platform travelled between the time of the laser measurement and the image capture (${}^{O_{t_2}}T_{LB}$ and ${}^{O_{t_1}}T_{RL}$ are reported in Table 2.2). Typical results are presented in Figure 2.10.



Figure 2.9: Typical panoramic image stitched together from the 5 cameras of the omnidirectional camera, mounted atop our research platform. To simplify the back projection of laser measurements we use only the single image that aligns closest with the field of view of the lasers (this also avoids unnecessary distortion of the images and visual artifacts such as those present along the grass verge in the image). The person in the image is standing at the interface between the “Campus” environment (the other side of the large wrought iron railings) and the “Parkland” environments. This entrance gate was only marginally wider than the research platform which led to a prolonged navigation surrounding this area. As discussed in Section 2.4.5 and shown in Figure 2.14, this navigation resulted in an accelerated accumulated error in trajectory estimation for this region.

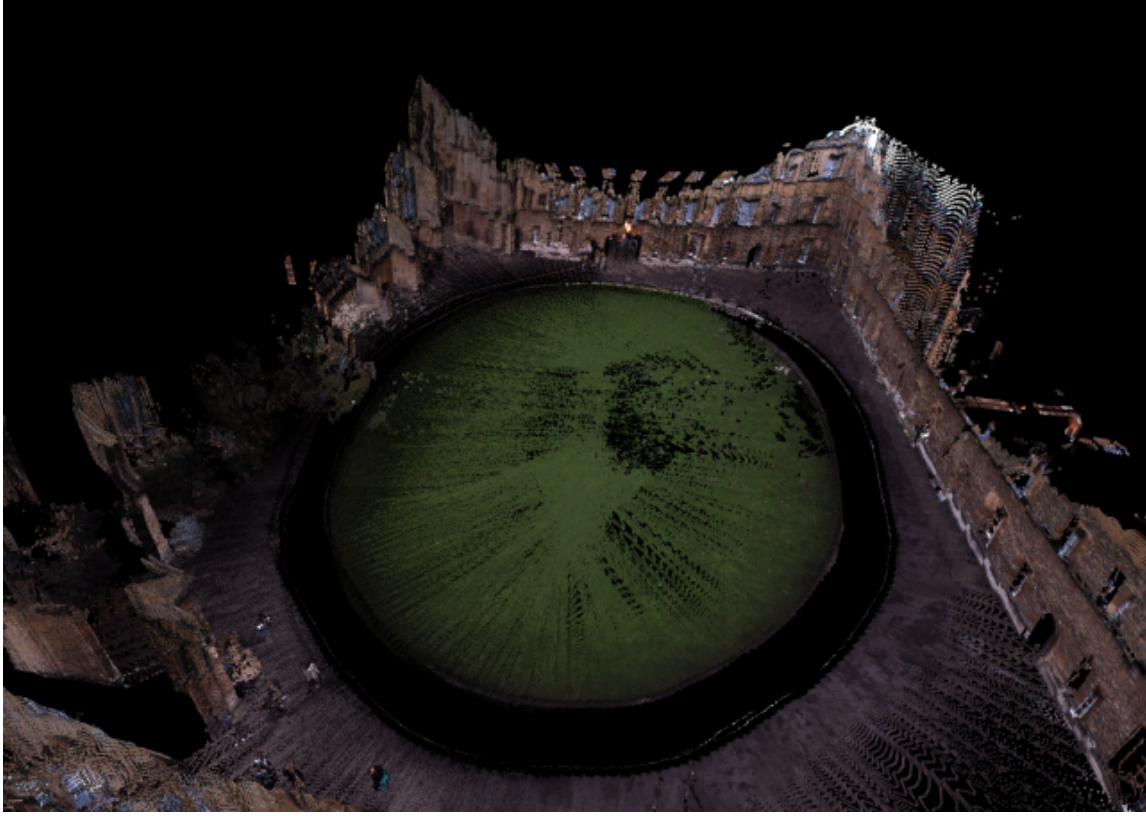


Figure 2.10: Elevated view of New College quad using coloured laser data (courtesy of Ingmar Posner). Colour information is produced through back projecting laser points into the view of the omnidirectional camera atop the research platform. As each sensor captures data at different times the relative transform between sensors necessarily involves an estimate of the vehicles trajectory. To minimise errors associated in both projection and trajectory estimates we use only the single images from the omnidirectional camera that align closest with the field of view of the laser scanner.

2.4.3 Visual Odometry

It is clear that the quality of the Segway odometry is insufficient if we wish to produce a globally accurate 3D point cloud of the complete dataset. GPS measurements are also intermittent and do not contain the full 6 DOF pose estimates required for the

construction of our point clouds. Instead we use Visual Odometry (VO) [64] to estimate the vehicle trajectory from images collected (at $20Hz$) from the forward facing stereoscopic camera. This technique tracks sparse, uniformly distributed, *fast corner* features of Rosten [91] that are present in the images. Independent temporal tracking of the left and right image sets provide a rough, rotation only (as translation per frame is small), initialisation of the pose estimate. This initialisation is used to provide consistency checks on features through back projecting both old features into the new frames, and between the left and right image pairs. Features are rejected if there are gross re-projection errors (using a sum of absolute difference error metric), or if they do not maintain the uniform sampling across the image area (quadtrees are used for speed of implementation). Full 6 DOF localisation is then applied to the remaining 50 – 100 features in each frame, using an m-estimator for robustness. It is important that the features are uniformly distributed across the images to ensure well conditioned transform estimations. When applied to our dataset we recover a 6 DOF trajectory estimate with (x, y) displayed in Figure 2.11.

2.4.4 Trajectory Relaxation using Visual Odometry, Place Recognition and Laser Data

Although these results are impressive compared to that of the Segway odometry, there is marked error between the final loop and the rest of the trajectory estimate. To correct this we use a technique called *loop closing* as detail in [69]. In essence loop closing attempts to reuse trajectory information in areas where the platform has revisited a location in the workspace multiple times. At these locations we can ascertain a

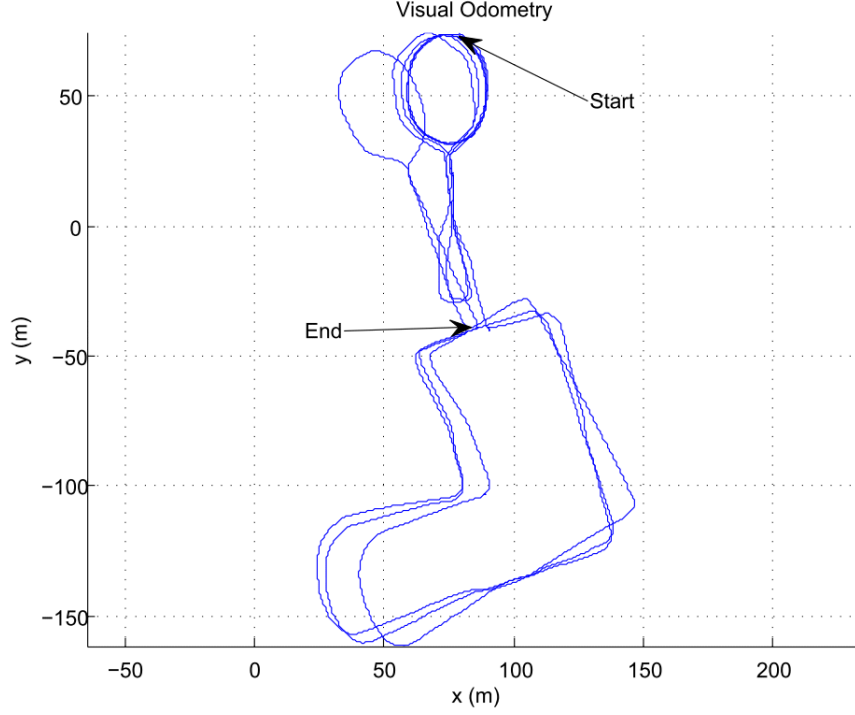


Figure 2.11: Research platform trajectory as estimated by Visual Odometry [64] for the New College dataset. There is small relative error in position over the 2.2km expedition when compared to the Segway odometry (Figure 2.5). The VO feature tracking algorithm performs well even in some cases where only moving leaves are visible in the scene.

relative transform between sections of the trajectory. Each transform is used as a loop constraint in the *relaxation* our trajectory estimate. Here the chain of 6 DOF vehicle pose estimates are combined with the loop closing constraints to form a pose graph. The edges of this graph are then perturbed to accommodate, in a minimum error sense, the metric information in all the constraints. The overall result of the optimisation is that errors are spread out along the entire trajectory. See Appendix B for a full treatment of the pose graph relaxation used in this work.

We use FAB-MAP (Fast Appearance Based Mapping) [23] to provide loop closure

locations based on the images from the omnidirectional camera. Using the appearance of locations alone allows loop closures to be robust to situations where there is significant error in the robot’s estimated metric position or even over multiple datasets. Significant error arises, for example, after traversing a large loop where turning angles have been poorly estimated. The approach is based on a Bayesian description of similarity and incorporates a generative model for typical place appearance. This allows the system to correctly assign loop closure probability to observations even in environments where many places have similar sensory appearance (a problem known as *perceptual aliasing*). The implementation is also computationally tractable through using a *bag-of-words* parameterisation (first suggested by [96]) and a Chow Liu tree that captures only first order conditional dependencies between appearance words. These dependencies attempt to embody the most important intrinsic associations between appearance words – for example, if a bike handle is found in an image then a saddle is far more likely to be found. They offer a marked improvement in results to those achieved using a naïve Bayes assumption while incurring little computational overhead.

With each of the loop closure locations identified we form estimates of the relative transformation between associated vehicle pose by using either stereo images or laser data. Where each pose of the loop closure is very similar we use the vision algorithm presented in Section 2.4.3 to estimate the transform between image pairs. Otherwise, and when this fails, a variant of the Iterative Closest Point (ICP) algorithm is used on the laser data that is gathered around the same time as the loop closure (see Appendix C for full details). In Figure 2.12a we see that laser registration is typically used when loop closure poses are from vehicle traversals in the opposite direction.

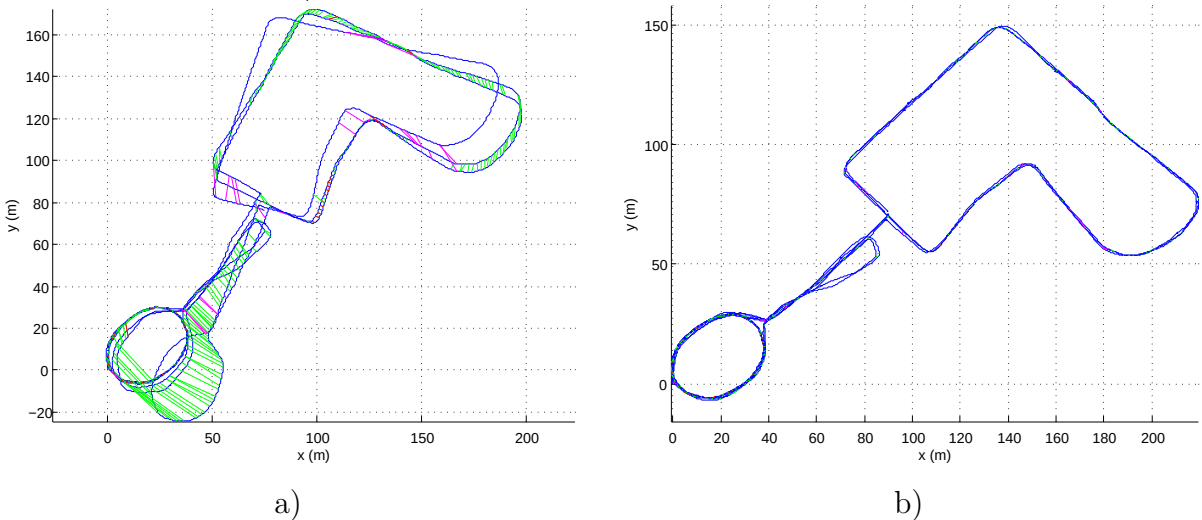


Figure 2.12: Visualisation of loop closure constraints used to relax Visual Odometry trajectory estimate for the New College dataset. (a) VO of Figure 2.11 with loop closure constraints overlaid. Green indicates constraints supplied through visual registration, while pink are laser registrations provided through our variant of the Iterative Closest Point (ICP) algorithm (detailed in Appendix C). (b) Depicts the final robot trajectory as estimated by our SLAM system.

The result of this fully automatic system is a highly accurate trajectory estimate as demonstrated in Figure 2.12b. We use this estimate for the rest of this thesis unless otherwise stated.

2.4.5 Laser Data using Relaxed Trajectory Estimates

Our improved pose estimates permit us to create visually compelling 3D laser point cloud as shown in the plan view of the complete New College dataset in Figure 2.13 and the view of the quadrangle that was in Chapter 1. At this scale the crispness of the features (such as the walls along the inside of the “dog leg” around the New College gardens - see Figure 2.13) indicate that our trajectory estimate are roughly consistent

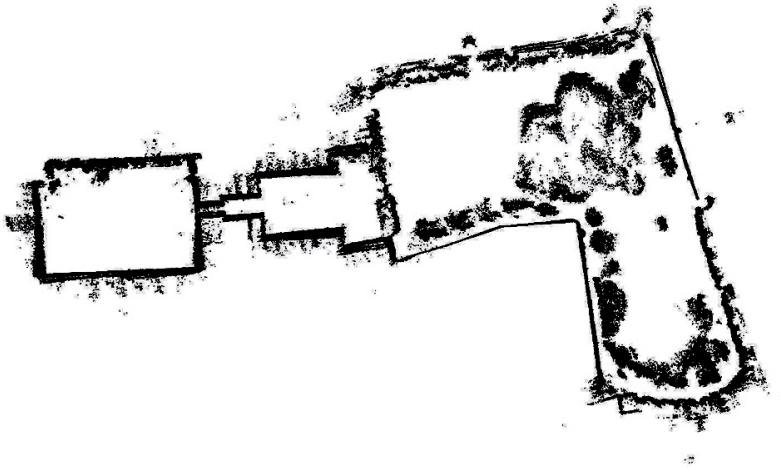


Figure 2.13: Plan view of laser data from the New College dataset using our optimised pose graph. Laser measurements of the ground have been removed so that prominent features of the workspace can be reconciled against the aerial photograph in Figure 2.3. The vast number of laser measurements (70×10^6 measurements) has required us to randomly downsample the measurements by a factor of 100 in order to render the entire dataset.

across the multiple circumnavigations of the test site.

We can visually check the extent that our data reflects the true geometry of the underlying workspace through manually overlaying our laser point cloud onto the aerial photograph of New College (Figure 2.14). If overlaid independently, the laser data for either the “Parkland” or “Campus” environments would highly correlate with the aerial view. The overall global mis-alignment can therefore be explained through poor trajectory estimates at the interface between these environments. Given the poor set of visual queues (see Figure 2.9) and prolonged navigation in this area this is justified, especially since the iron screening at this interface effectively partitions the workspaces. The result is that there are no cross-environment loop closures (see Figure 2.12) which would have otherwise constrained our pose graph optimisation and corrected the error



Figure 2.14: Ariel view of the New College test site with laser measurements superimposed. The point cloud was manually stretched and rotated to fit the aerial view based on the vertical structures that appear in both representations of the workspace. We see that overall there is a high level of agreement between representations, bar a rotational error in trajectory estimation surrounding the interface between “Campus” and “Parkland” environments near the centre of the figure. This led to an angular error of the “Parkland” point cloud relative to the overhead image.

present in the overlay of Figure 2.14.

This discussion has identified macro level errors in our laser point clouds. We return to map quality assessment in Chapter 3 with a more formal treatment that analyses the level of alignment of micro sections of laser data that are generated from distinct passes of the same workspace. Both investigations are then used to inform our laser data parameterisation which we use throughout this dissertation.

Chapter 3

Representations Of Laser Data For Push Broom Laser Systems

The work on assessing map quality presented in this chapter was published in ‘Navigating, Recognising and Describing Urban Spaces with Vision and Laser’ in the International Journal of Robotics Research, 2009. Our parameterisation was first published in ‘Efficient Non-Parametric Surface Representations Using Active Sampling for Push Broom Laser Data’ in the Proceedings of Robotics: Science and Systems, 2010.

In this chapter we introduce our parameterisation of data that is collected by a push broom laser system such as the one discussed in Chapter 2. This parameterisation is used throughout this dissertation. It is guided by an analysis of the global accuracy of push broom laser point clouds and the need for an accurate representation of the general underlying workspace surfaces that are captured in laser data.

Motivated by our analysis in Section 3.2 we move away from performing the regression of laser data in a global frame of reference and instead regress in what we term *sensor configuration space*. To the best of our knowledge, this parameterisation is novel, and it provides several benefits as we shall discuss in Section 3.3. We then discuss the characteristics that our laser data has in this sensor configuration space in Section 3.4. Finally, we provide a method to assess the accuracy of predictions of the underlying workspace surfaces in Section 3.5 which we shall discuss in subsequent chapters.

3.1 New Nomenclature Introduced in this Chapter

Laser sensor model

- $\mathcal{W}$ - 3D Euclidean workspace
- $\mathbf{x}$ - a single point in 3D Euclidean workspace $\mathbf{x} \in \mathcal{W}$
- $\mathcal{Q}$ - sensor configuration space
- θ - angular position of laser beam relative to the laser sensor
- t - timestamp of laser range measurement
- $\mathbf{q}$ - sensor configuration $\mathbf{q} = [\theta, t]^T$ in sensor configuration space $\mathbf{q} \in \mathcal{Q}$
- ψ_θ - angular scale factor so that laser measurements form a unit spaced grid in $\mathcal{Q}$ ($\psi_\theta = 0.5^\circ$ – the angular resolution of the scanner)
- ψ_t - times scale factor so that laser measurements form a unit spaced grid in $\mathcal{Q}$ ($\psi_t = \frac{1}{75}$ seconds – the temporal resolution of the scanner)
- $\mathbf{p}$ - 6 DOF pose of the sensor's laser beam
- N - number of laser measurements in a laser data dataset
- $\mathbf{r}$ - all laser range measurements in a dataset $\mathbf{r} = [r_1, r_2 \dots r_N]^T$
- $\mathcal{Q}$ - set of all sensor configurations in a dataset (training data) $\mathcal{Q} = \{\mathbf{q}_i\}_{i=1}^N$
- $\mathcal{D}$ - set of all laser measurements $\mathcal{D} = \{(\mathbf{q}_i, r_i)\}_{i=1}^N$
- $G(.)$ - mapping that maps a sensor configuration $\mathbf{q}$ to a range measurement r
- $V(.)$ - mapping that maps a sensor configuration $\mathbf{q}$ to a vehicle pose $\mathbf{p}$
- $E(.)$ - mapping that maps a vehicle pose $\mathbf{p}$ and a laser range measurement r to a 3D point $\mathbf{x}$ in Euclidean workspace $\mathcal{W}$

3.2 Global Representation Using Pose Estimates

In Section 2.4 we generated 3D laser point clouds by trawling a 2D laser scanning system through space on a dynamically balanced mobile robot (see Figure 2.6). Laser range and bearing measurements were projected into a local 2D sensor frame. These measurements were then further projected, first through a fixed transform to the local vehicle frame, and then into the 3D world frame through the pose of the vehicle at the time that each measurement was made. As discussed, vehicle poses were supplied through either wheel odometry and inertial measurements, or estimated from Visual Odometry and refined through automatic pose relaxation (see Appendix B). This refinement exploited metric and topological constraints in an attempt to form a globally accurate metric map as in [69]. The resulting point clouds had a very high *local consistency* (you can read the car number plate in Figure 2.8 for example). by using the latter technique we also generated a visually compelling 3D point Cloud over the complete 2.2 km trajectory which embodied several circumnavigations of the New College dataset test site (Figure 2.13), despite a large range of motion in our platform. However, even with pose relaxation, we found that small errors at pathological locations along the robot’s trajectory could have devastating affects on the global accuracy of laser point clouds, as seen in Figure 2.14.

Above and beyond representing laser data as a point cloud, however, our work aims to capture the underlying workspace *surfaces* as accurately and efficiently as possible. In the formation of our framework, we first consider the *global consistency* of our laser point cloud. Global consistency is concerned with how well measurements between multiple observations of the same region of the workspace align in the global frame (this is in

contrast to *local consistency* which concerns only a single pass of the workspace which is highly consistent for our laser data, as shown in Figure 2.8). This consideration is important in deciding if observations are sufficiently aligned to permit a single highly accurate surface representation (for example by using a single privileged global frame to represent all our data); or if we must find a representation that can allow for larger discrepancies.

The following analysis guided the creation of our mapping representation which we introduce in Section 3.3 and provides a baseline for the evaluation of our system against future work.

3.2.1 Assessing Global Map Consistency

We can begin to appreciate the importance of global consistency through a qualitative examination of the laser data we presented in Chapter 2. Birds-eye renderings of laser data for the New College quad are depicted pre-pose relaxation in Figure 3.1 and post-pose relaxation in Figure 3.2 (relaxation was performed over the entire 2.2 km trajectory of the dataset using the approach outlined in Section 2.4.4). Although the local consistency of the point cloud has degraded slightly as a result of the pose relaxation (see the curving of the walls when Figure 3.1a is compared against Figure 3.2a), global consistency has increased dramatically as distinct observations of the walls in Figure 3.1b are aligned in 3.2b.

For a quantitative analysis of global consistency we consider the map built from our SLAM system (see Chapter 2) pre pose relaxation (see Section 2.4.4). Our approach is to measure how far from ideal our map is by calculating statistics on the consistency of

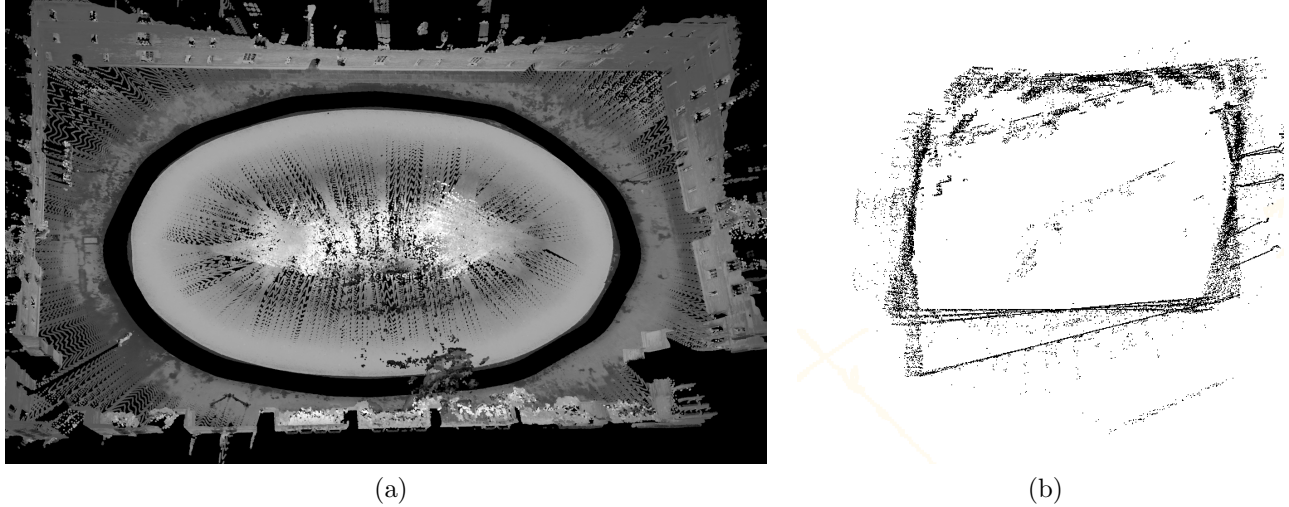


Figure 3.1: Birds-eye views of a 3D laser point cloud of the New College quadrangle without using pose relaxation to refine vehicle trajectory estimates (see Chapter 2 for detailed information). (a) depicts all laser measurements for one circumnavigation of the quad. Adjacent quadrangle walls are perpendicular to each other and highly planar, demonstrating a high local consistency. (b) shows a downsampled view of laser data that was collected across all 6 circumnavigations of the quadrangle in the dataset. Although inaccuracies in our Visual Odometry algorithm are small these can accumulate over the 2.2km trajectory estimate and lead to grossly globally inconsistent point clouds. For example, one can clearly distinguish that multiple observations of the same planar walls sections are spatially separated by a significant amount towards the lower left of the figure.

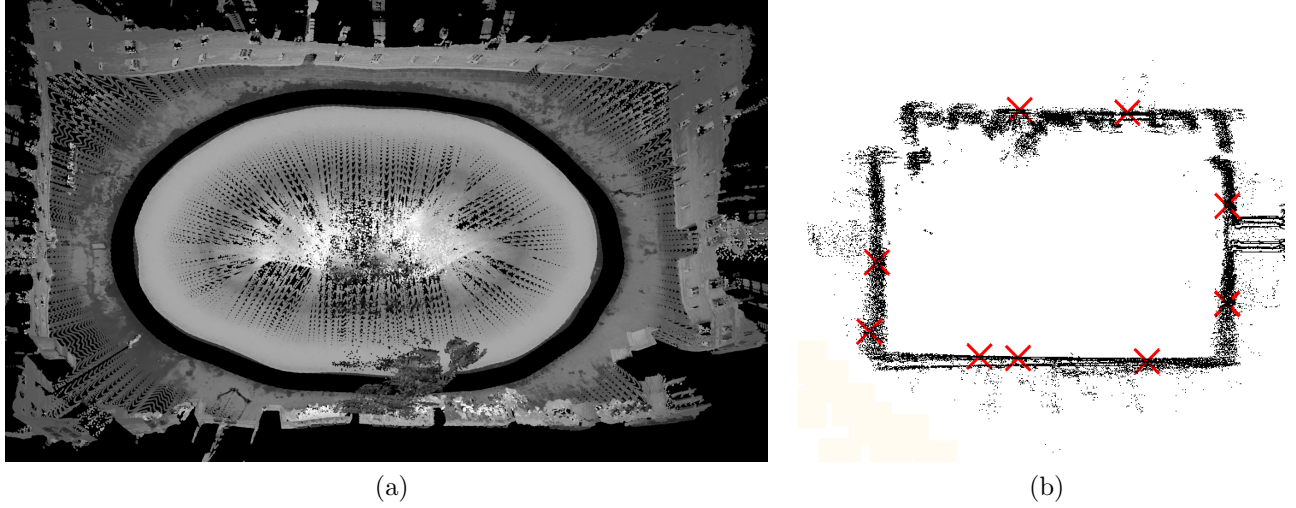


Figure 3.2: Birds-eye views of a 3D laser point cloud of the New College quadrangle using pose relaxation to refine vehicle trajectory estimates along the entire 2.2 km trajectory (see Chapter 2 for detailed information). (a) depicts all laser measurements for one circumnavigation of the quadrangle. There is significant curving of the walls along the top and bottom of these images, and thus a decrease in local consistency when compared to Figure 3.1a. This is due to our pose relaxation algorithm which optimises our trajectory estimate constrained on all topological and metric constraints (see Section 2.4.4). (b) shows a downsampled view of laser data that was collected across all 6 circumnavigations of the quadrangle in the dataset. In comparison with Figure 3.1b we see that pose relaxation has had a dramatic positive effect on the global consistency to such an extent that we may now apply our simple quantitative analysis of Section 3.2.1. With reference to Section 3.2.1, red crosses depict the L locations of the user generated and accepted seed points that are used to analyse the extent of global point cloud inconsistency in this scene.

repeat observations of multiple sections of wall within the workspace. These statistics are semi-manually captured for each re-observations of each of the wall sections. They are computed by finding a planar set of points for each observation. We call these *capture point patches* and generate them by following three steps:

Region of interest selection. The user is presented with a 3D point cloud of the *initial* pass of the New College quadrangle and selects K seed points ${}^1p_{1:K}$. A capture radius r_i is expanded around each 1p_i to select all neighbouring capture point patches ${}^1\mathcal{C}_i$ within the sphere it defines (Figure 3.3). Here we are using the superscripted prefix to indicate the pass of the workspace – 1 being the first pass, 2 being the second and so on.

Cluster generation. A large search sphere (of the order of metres) is used to find the K sets of ${}^{2:M}p_i$ which correspond to the user selected seed points 1p_i from each of the $M - 1$ subsequent passes of the workspace. Correspondence is determined through the selection of points that have the lowest Euclidean distance to the seed points and that have distinct timestamps to each other. This ensures that each point that is selected belongs to a distinct passes of the same location, as required. We then find each of the capture point patch sets ${}^{2:M}\mathcal{C}_i \quad i = 1 : K$ through additional sphere searches using corresponding r_i as found in the previous step. We refer to the set of all capture point patches for a given seed point 1p_i , ${}^{1:M}\mathcal{C}_i$ as a *cluster*. This simple sphere search technique assumes that there is sufficient global consistency whereby all wall positions on each pass in the map are not in gross error, as is the case our post-relaxation data (see Figure 3.2b).

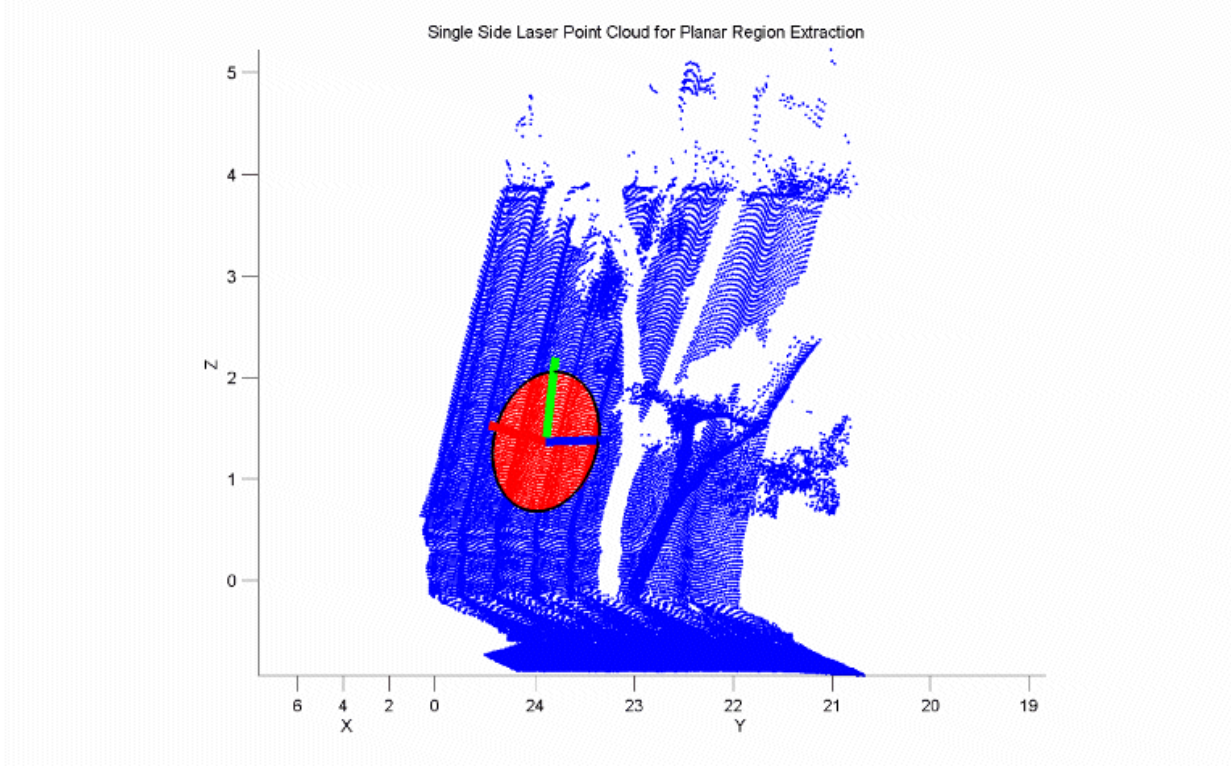


Figure 3.3: Example of a user selected capture point patch for a section of 3D laser data from the New College quadrangle. The user selected seed point 1p_i with corresponding capture point patch ${}^1\mathcal{C}_i$ is highlighted in red. The extent of capture radius r_i is depicted as a black circle with its principle axes aligned to the capture point patch.

If this was not true then finding correspondences across passes would be difficult. We are also interested in only the closest sections of the wall according to our map for each pass, and do not require that each of the ${}^{2:M}p_i$ are not in exactly the same location on the underlying wall (this would be virtually impossible given that laser samples represent measurements at only discrete locations). Instead we ensure that each of the capture point patches in a cluster ${}^{1:M}\mathcal{C}_i$ are of roughly the same wall section by performing the following step.

Cluster rejection. For each of the K seed points, ${}^{1:M}\mathcal{C}_i$ capture point patches are rejected if any of the $2 : M$ planar regions of interest do not lie completely on a wall, leaving a total of L accepted seed points ${}^1p_{1:L}$ and corresponding clusters ${}^{1:M}\mathcal{C}_i \quad i = 1 : L$. We do this as our metrics are only interested in the alignments of corresponding **planar** sections across different passes. The red crosses in Figure 3.2b show the final L accepted seed points used in this evaluation which cover varying positions along all four quadrangle walls.

We are now able to calculate our consistency statistics. Firstly we calculate the normal ${}^j\hat{\mathbf{n}}_i$ of each wall patch ${}^j\mathcal{C}_i$ via a Singular Value Decomposition of its scatter matrix and also the centroids ${}^j\mathbf{c}_i$, $j = 1 : M \quad i = 1 : L$. For each of the N possible pairing across the M planes within the same cluster, we calculate the Euclidean distance between centroids $\|\cdot\|_{j=1:M}$ and the angle between the surface normals $\angle_{j=1:M}(\cdot)$. These distances, coupled with the normal information, encapsulates everything we need to know for our global consistency analysis, and we refer to these quantities as

$$\text{intra-cluster displacement: } \boldsymbol{\alpha}_i = \left\| {}^j\mathcal{C}_i \right\|_{j=1:M} \quad (3.1)$$

$$\text{intra-cluster alignment: } \boldsymbol{\beta}_i = \angle_{j=1:M}({}^j\mathcal{C}_i) \quad (3.2)$$

where each quantity has as many elements as there are pairings N . We could also calculate secondary statistics such as patch curvature or spread to check local consistency. However, at the scale of the patches that can be selected, these secondary metrics are dominated by noise from the variation in the shape of the underlying walls (they are not perfectly flat and uniform). Table 3.1 presents statistics on our quantities

Property	Value (Post-relaxation)
i) max. intra-cluster displacement: $\max \{\alpha_i, i = 1:L\}$	0.6 m
ii) min. intra-cluster displacement: $\min \{\alpha_i, i = 1:L\}$	0.02 m
iii) max. of the ave. intra-cluster displacement: $\max \{\text{mean} \{\alpha_i\}, i = 1:L\}$	0.33 m
iv) min. of the ave. intra-cluster displacement: $\min \{\text{mean} \{\alpha_i\}, i = 1:L\}$	0.14 m
v) ave. of the ave. intra-cluster displacement: $\text{mean} \{\text{mean} \{\alpha_i\}, i = 1:L\}$	0.21 m
vi) max. intra-cluster angle: $\max \{\beta_i, i = 1:L\}$	9.1°
vii) min. intra-cluster angle: $\min \{\beta_i, i = 1:L\}$	0.32°
viii) max. of the ave. intra-cluster angle: $\max \{\text{mean} \{\beta_i\}, i = 1:L\}$	4.86°
ix) min. of the ave. intra-cluster angle: $\min \{\text{mean} \{\beta_i\}, i = 1:L\}$	0.66°
x) ave. of the ave. intra-cluster angle: $\text{mean} \{\text{mean} \{\beta_i\}, i = 1:L\}$	3.6°

Table 3.1: Quantitative analysis of the global consistency of the New College quadrangle laser data. Trajectory estimates were provided through Visual Odometry which was refined using pose relaxation that used metric and topological constraints along the entire 2.2km trajectory of the dataset (see Section 2.4.4). Statistics are collected for all N possible pairings of clusters ${}^{1:M}\mathcal{C}_i$ that correspond to each of the L accepted seed points (depicted in Figure 3.2b).

across all of the accept seed points $i = 1 : L$.

From the table we see that the minimum intra-cluster displacement (Equation 3.1) across all selected wall sections was 0.02 m. This minimum was achieved between neighbouring circumnavigations of the quadrangle over a trajectory of roughly 150 m. When taking into account that the precision of the laser scanner is 0.015 m this a promising result that demonstrates the accuracy of our SLAM system. In the context of the size of the New College quadrangle (40×55 m) the maximum intra-cluster displacement 0.6m is also small, and a large improvement over results that could be obtained if pose

relaxation was not used (in Figure 3.1b we see that the spread of the walls is into the metre range). However this maximum is an order of magnitude greater than the measurement precision and is hence significant. In addition, the next set of statistics (properties (iii) through (v) in Table 3.1) loosely indicates a high perpendicular spread of each cluster, measured by the average distance between centroids across all pairings. We therefore believe that in order to produce a single accurate surface representation one would require a representation that explicitly accounts for distinct passes of the same wall sections.

We use intra-cluster alignment (Equation 3.2) statistics to reflect the amount that the New College quadrangle walls have been bent in order for the vehicle trajectory to accomodate our relaxation constraints. We can do so by acknowledging that the quadrangle is predominantly constructed from 4 planar walls that intersect at right angles (see the overhead image of Figure 2.3) and that this is evident in our laser data pre-pose relaxation (see Figure 3.1a). Since all 4 walls roughly align in our post relaxation data, deviations from this oblong shape must therefore be largely due to our relaxation algorithm. Despite only visibly small undulations in the walls of Figure 3.2a we found that in the worst case two re-observations of the wall section were at 9.1° from each other. Further, the average miss-alignment across each cluster ranged between 0.66° and 4.86° . This means that even when there is a low intra-cluster displacement (Equation 3.1) sometimes these displacements could increase dramatically away from the centroid of the wall sections.

3.2.2 Discussion

In our analysis we sought to gauge the amount of global consistency that is present in our current SLAM system [69]. In doing so we used a single representative dataset that was generated from our robotics platform (Chapter 2). We used basic statistics which we collected for the simplest of surfaces that are present in our point clouds - that of a plane. The results will be used to provide a baseline for future evaluations and they have guided us in the creation of a mapping representation that is robust to the typical alignment errors that are associated with these inconsistencies as we shall discuss in the following section.

For ease of analysis we restricted the creation of seed points to the highly planar quadrangle of New College. The analysis provided insight over the entire dataset however, as the pose relaxation in our system was employed using all constraints found throughout the trajectory (see Section 2.4.4). This was important as we wanted to gather robust information about our system in a variety of different scenes in an urban environment, including parkland, and in the presence of small errors in trajectory estimates (in this dataset this occurred at the intersect between two distinct workspace areas as noted in Section 3.2).

We found that the effect of small trajectory errors were manifested in the poor alignment of quadrangle walls in the pre-relaxed point cloud of Figure 3.1b. We then further ascertained the extent that these errors had been mitigated through pose relaxation (Figure 3.2).

Our results indicated that there are some areas of our laser data with high global consistency. In these areas each re-observation of the same underlying surface were in

very close proximity in the global frame. This is a promising result as one can imagine combining multiple re-observations simply by processing all the laser measurements in these areas together. A single unified surface representation could then be generated through, for example, parametric curve fitting. One can envisage an approach that uses our simple statistics, combined with randomised seed points, to automatically identify regions in the dataset where a unified representation is feasible. We may also be able to add extra constraints to the pose relaxation as a function of the measured quality of the maps (Equations 3.1 and 3.2). As trajectory estimates become more accurate we would expect that the proportion of highly globally consistent laser data will increase, and more sophisticated secondary statistics may become more viable in the above analysis. Localisation systems that incorporate pose refinement based on available global information sources such as [65, 51] may alternatively provide sufficiently high global consistency. [65] for example uses overhead imagery to constrain trajectory estimates to the road network for use in autonomous road vehicle applications.

However even with our state-of-the-art SLAM system we found that in many of our wall sections the displacement between re-observations were significant. The analysis also indicated considerable bending of the New College quadrangle walls which were as a result of inconsistencies between our relaxation constraints. In addition, although significant errors are rare there will always be times when parts of our system break, for example, when point cloud and vision registration algorithms provide estimates that are inconsistent (for example, due to a changing workspace), there is a failure to recognise a “loop closure” (when the platform has returned to the same location - see Section 2.4.4), or even when our vision system undergoes catastrophic failure (for

example, when a falling leaf fills the majority of the cameras field of view). It is not clear how multiple observations of the same workspace can be accurately reconciled into one unified surface that is robust under these conditions.

In the following section we introduce a parameterisation of our laser data that effectively side-steps the challenges involved in producing a unified surface representation under poor global consistency. As we shall discuss, these parameterisations permits us to arbitrarily change our vehicle trajectory estimates with no computational overhead on our mapping algorithm.

3.3 Sensor Configuration Space parameterisation

We require that a successful parameterisation of our laser data must be capable of representing underlying surfaces that are present in the workspace while addressing the problems associated with a single global frame of reference as discussed above. Further, the workspaces that we are interested in have heterogeneous geometric properties and are sampled at varying densities. As we shall discuss in Chapter 4, Gaussian Processes provide a very flexible technique for modelling this workspace as they can represent surfaces with arbitrary complexity and provide access to uncertainty which reflects the varying sampling density. They have been used successfully in terrain modelling applications to model a function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ which associates the range as a single elevation value z and a domain of any given position in (x, y) of 3D Euclidean space as discussed in the literature review in Section 1.2. However, while this application of GPs is effective for terrain modelling it precludes application in the full 3D mapping

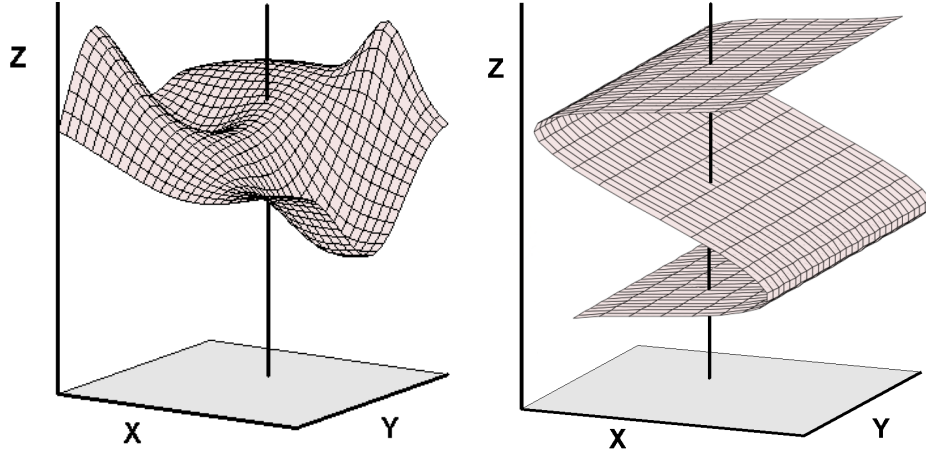


Figure 3.4: Visualisation of the requirements of terrain mapping versus general 3D mapping. We wish to map scenes with a non-functional relation to the ground plane – that is multiple elevations z for a given x, y point (right), rather than being restricted to trivial, well-formed functions (left).

case. That is, our parameterisation must be able to represent common structures such as walls which require multiple elevation values to be associated with the same position on the x, y -plane. Technically, this multivalued *relation* is *non-functional*, and we use this term to distinguish it from that of a single valued, *well-formed* function (see Figure 3.4 for an example of both these relations in the context of mapping).

Our approach admits the representation of any (x, y, z) point that lies on the sampled workspace surfaces by reforming this regression problem to exploit the nature of push broom laser data. In particular, we pose these queries in (x, y, z) as laser range queries along arbitrary rays emanating from the sensor over a domain that represents a point

along the sensor’s trajectory. Let a beam from the laser be parametrised by a point $\mathbf{q} \in \mathbb{S} \times \mathbb{R}$ where $\mathbf{q} = [\theta, t]^T$. For a push broom laser θ denotes the 1D angular position of the laser beam (*not the whole sensor unit*) and t denotes the timestamp of the laser scan.¹ We describe $\mathbf{q}$ as a point in *sensor configuration space* $\mathcal{Q}$, as illustrated in Figure 3.5, and refer to it as a particular configuration of the laser sensor. We note that this representation is not bound to the push broom laser system, as additional dimensions in θ can be used to reflect, for example, additional scanning dimensions that are present in other laser scanner such as the HDL-64E by Velodyne or flash LIDAR systems (see Section 8.1.3 for further discussion).

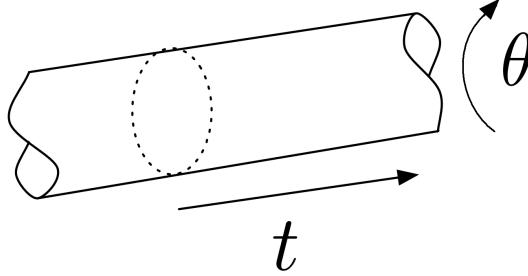


Figure 3.5: Illustration of sensor configuration space $\mathcal{Q}$ for a push broom laser data. Every point $\mathbf{q}$ on this manifold is parametrised by time t and the angular position of the laser beam θ .

Sensor configuration space provides a natural domain for laser range regression since it is closely related to the state of the laser and requires no information about the rest of the robotic system. The mapping G from sensor configuration space to range is necessarily a well-formed function – our laser sensor associates a single range r with

¹For our robotics system θ refers to the angle from vertical since our laser scanners are mounted so that their scan plans are vertically aligned, perpendicular to the direction of travel of the vehicle.

any point in $\mathcal{Q}$ such that

$$G : \mathbb{S} \times \mathbb{R} \rightarrow \mathbb{R}, \mathbf{q} \mapsto r \quad (3.3)$$

Consider now a mapping from $\mathcal{Q}$ to the Euclidean workspace $\mathcal{W}$

$$E'(\mathbf{q}) = E(V(\mathbf{q}), G(\mathbf{q})) \quad (3.4)$$

where:

$$V : \mathbb{S} \times \mathbb{R} \rightarrow \mathbb{S}^3 \times \mathbb{R}^3, \mathbf{q} \mapsto \mathbf{p} \quad (3.5)$$

$$E : (\mathbb{S}^3 \times \mathbb{R}^3) \times \mathbb{R} \rightarrow \mathbb{R}^3, (\mathbf{p}, r) \mapsto \mathbf{x} \quad (3.6)$$

For every point $\mathbf{q}$ in $\mathcal{Q}$, $V(\mathbf{q})$ provides the 6 DOF pose of the sensor's laser beam $\mathbf{p} \in \mathbb{S}^3 \times \mathbb{R}^3$ at the time a measurement is taken – represented as roll, pitch, yaw and position. $E(\mathbf{p}, r)$ maps a 6 DOF laser beam pose $\mathbf{p}$ and a scalar range measurement r to a single point $\mathbf{x}$ in 3D Euclidean space $\mathcal{W}$. This parameterisation therefore naturally eschews the non-functional relation between elevation z and a domain of (x, y) location.

By ensuring that each operation is distinct (Equations 3.3, 3.5 and 3.6), we decouple robot trajectory estimation $V(\mathbf{q})$, from that of the regression of the laser data $G(\mathbf{q})$.² There is therefore no computational overhead associated with our mapping algorithm for arbitrary changes in $V(\mathbf{q})$. This is particularly beneficial for online applications where $V(\mathbf{q})$ often changes especially under the presence of loop closure information and

²Specifically $G(\mathbf{q})$ is independent of the **estimation** of $V(\mathbf{q})$ but not the ground-truth $V(\mathbf{q})$ of the sensor; any laser range measurements of the workspace will always be dependent on the relative position of the sensor and the workspace surfaces.

pose relaxation (see Section 2.4.4 for a discussion of both).

In the following section we analyse the implications that this parameterisation has on our regression algorithm.

3.4 Laser Data in Sensor Configuration Space

Using our parameterisation we can now pose the problem of surface modelling as that of performing a regression over range in sensor configuration space $\mathcal{Q}$. To understand the properties that laser data exhibits in $\mathcal{Q}$, we use the example of a tree (Figure 3.6) from the North wall of the New College quadrangle that we first saw in Figure 3.3, and compare it to the same data in Euclidean space $\mathcal{W}$.

Our push broom laser system comprises of a laser scanner with its scan plane aligned vertically in $\mathcal{W}$ (see Chapter 2.1.1). For this specific data collection example this configuration produced a sparse, non-uniform, vertical stripping pattern that is prominent in the lower half of the Euclidean plot of the laser data (Figure 3.6b). It is the combined angular resolution of the scanner and the proximity of the scanner to the workspace surfaces (the floor near the sensor) that created a relatively high in-scan-plane resolution in this region compared to the out-of-scan-plane resolution that is due to the distance that the sensor travelled between scans. More generally, the velocity of the sensor (in particular changes in the sensor’s yaw and pitch) as it progresses along its trajectory affect the out-of-scan-plane resolution but for this example the sensor travelled roughly in a straight line in front of the wall. Vertical resolution increases with scan angle in $\mathcal{W}$ as the laser traces out measurements away from the sensor and up the quadrangle

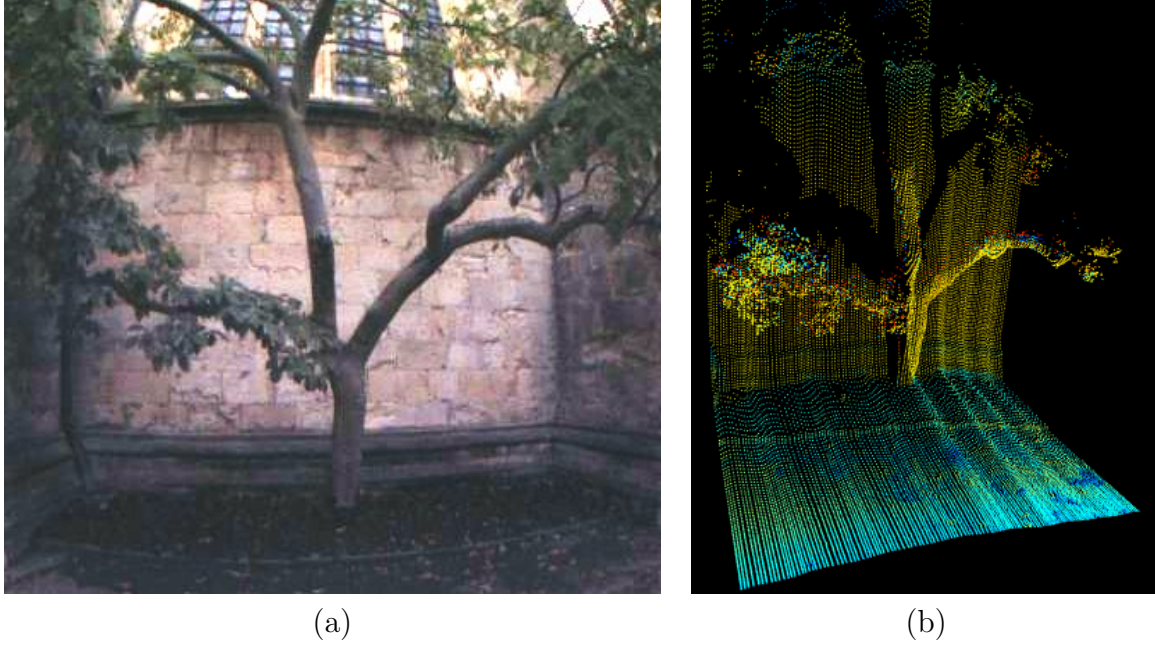


Figure 3.6: Example scene of a tree from the New College dataset [97]. (b) visualisation of the laser data in Euclidean space that was collected roughly at the same time as the image in (a). The visualisation was constructed using trajectory estimates provided by our Visual Odometry system (as discussed in Section 2.4.4). The sensor progressed roughly parallel to the backing wall from left to right in the image. This example is used throughout this section starting with a detail view of the tree trunk area in Figure 3.7 and subsequent visualisations of this tree trunk in sensor configuration space in Figure 3.9 and later in Figure 5.1. Colouring of the laser points in the visualisation is based on reflectance of each laser point as detailed in Chapter 2 and is used for only aesthetic purposes.

wall. Figure 3.7 demonstrates how the rolling motion of the sensor is also embodied in the positioning of the laser measurements in $\mathcal{W}$. A surface has been generated through forming a triangular mesh from the laser measurements for just the tree trunk and surrounding wall of Figure 3.6. The scan data is bounded in scan angle θ between 80° and 110° from vertically downwards and one can clearly observe waviness along these boundaries which are due to the motion of the data capture vehicle. For a visualisation of the data as it is captured by our robotics system see Figure 3.8.

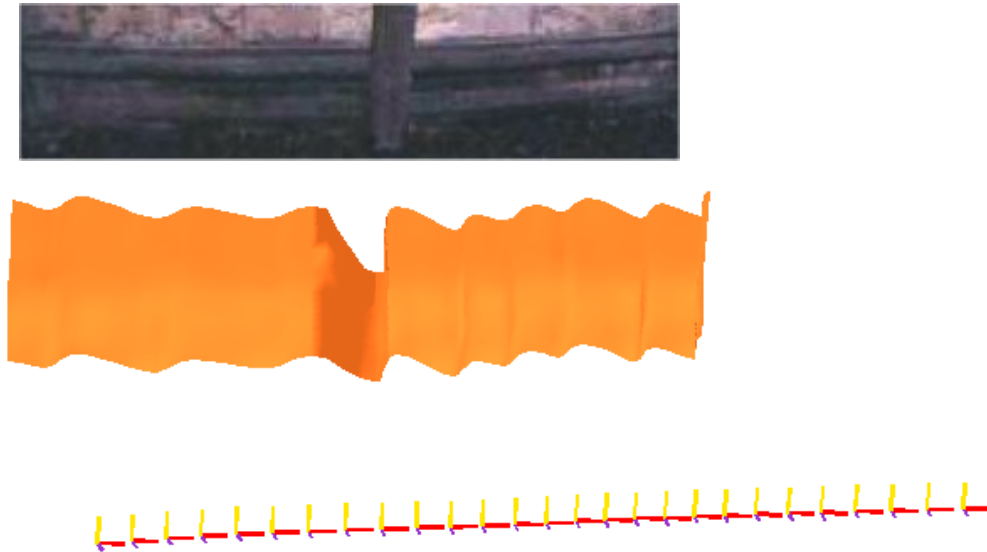


Figure 3.7: Tree trunk section of Figure 3.6 taken from the New College dataset [97]. (Bottom) is a visualisation using only laser data bounded between 80° and 110° from vertically downwards in Euclidean space that was collected roughly at the same time as the image in (top). The visualisation was constructed using trajectory estimates provided by our Visual Odometry system (as discussed in Chapter 2) and then a surface was drawn from a triangularisation of the resulting laser points. Coloured coordinate frames across the bottom of the visualisation indicate the trajectory of the vehicle that was involved in data capture (the laser sensor was mounted above and slightly to the left of the vehicle frame as shown in Figure 2.2 of Chapter 2). The large crease in the surface is due to the tree trunk which occludes the wall in the background of the image.

In all of our sensor configuration space $\mathcal{Q}$ visualisations we use the timestamp and angular position of the laser beam relative to the sensor frame as the horizontal axes, (corresponding to t and θ of Section 3.3), and the sensor’s range measurements as the vertical axis (r in Section 3.3) in a 3D plot. Each range measurement is therefore represented as a single 3D point. The well-formed functional relation between the x , y and z of our measurements in Euclidean Space $\mathcal{W}$ is manifested by all of our regression surfaces in this space being strictly 2.5 dimensional – only a single height r is associated with any position in the domain θ , t .

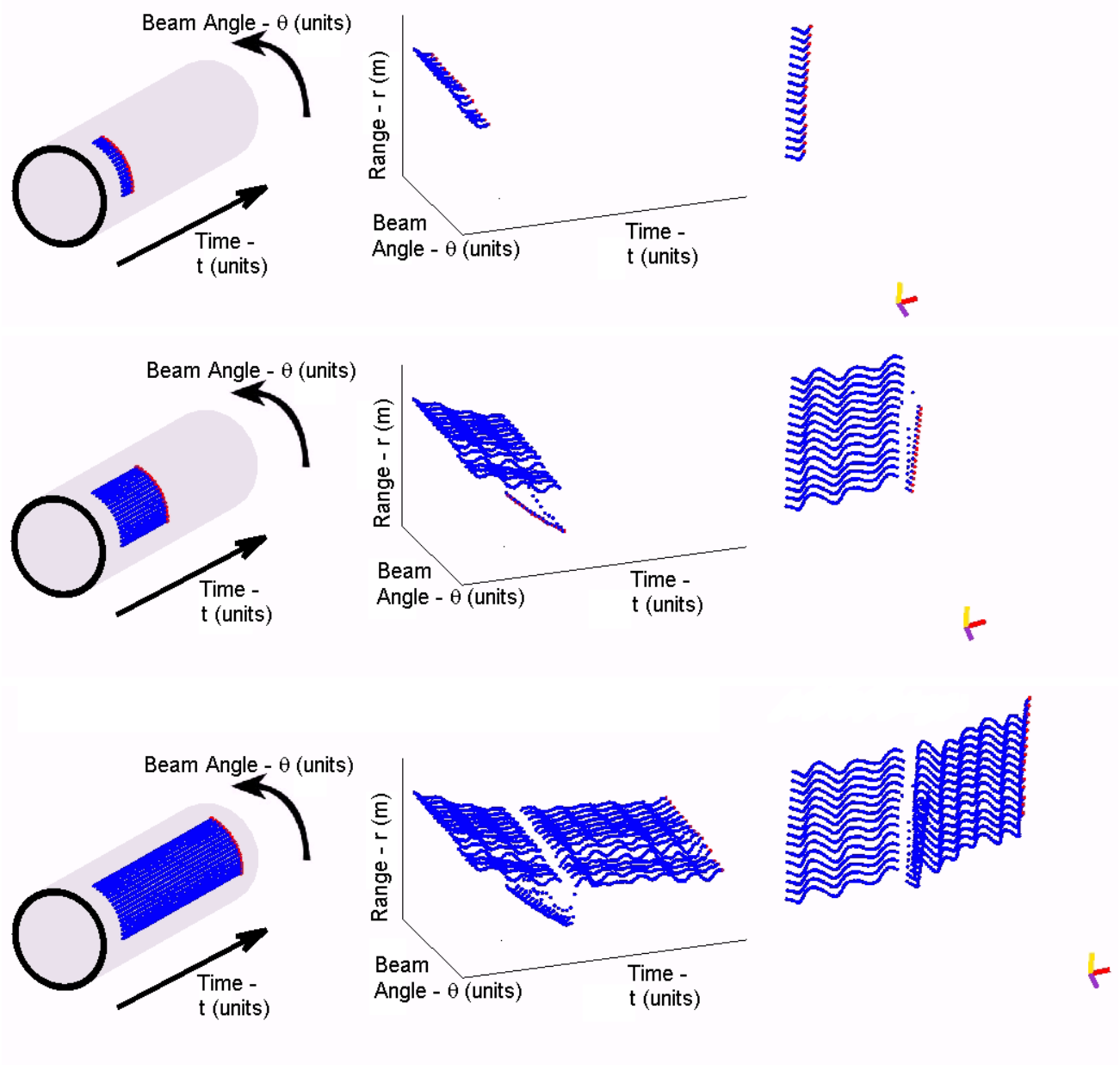


Figure 3.8: Timeseries visualisation of the tree trunk example of Figure 3.7 taken from the New College dataset [97] (see Chapter 2 for a detailed account of the robotics research platform). Each row represents progressive snapshots in time. Red points indicate the most recently captured measurements, while blue are the remaining measurements that have been captured since the beginning of the example. The left column indicates the sensor configuration $\mathcal{Q}$ visualisation of Figure 3.5. The middle column unwraps $\mathcal{Q}$ to span the horizontal axes and has the output space of the regression, range $\mathbf{r}$ on the vertical axis. The right column has the same laser data but projected into Euclidean space $\mathcal{W}$ through an estimate of the vehicle trajectory (supplied by the algorithm discussed in Section 2.4.4). The vehicle pose at the time of the snapshot is visualised by the coloured coordinate frame across the bottom of the plot. For a detailed description of the data see Figure 3.9.

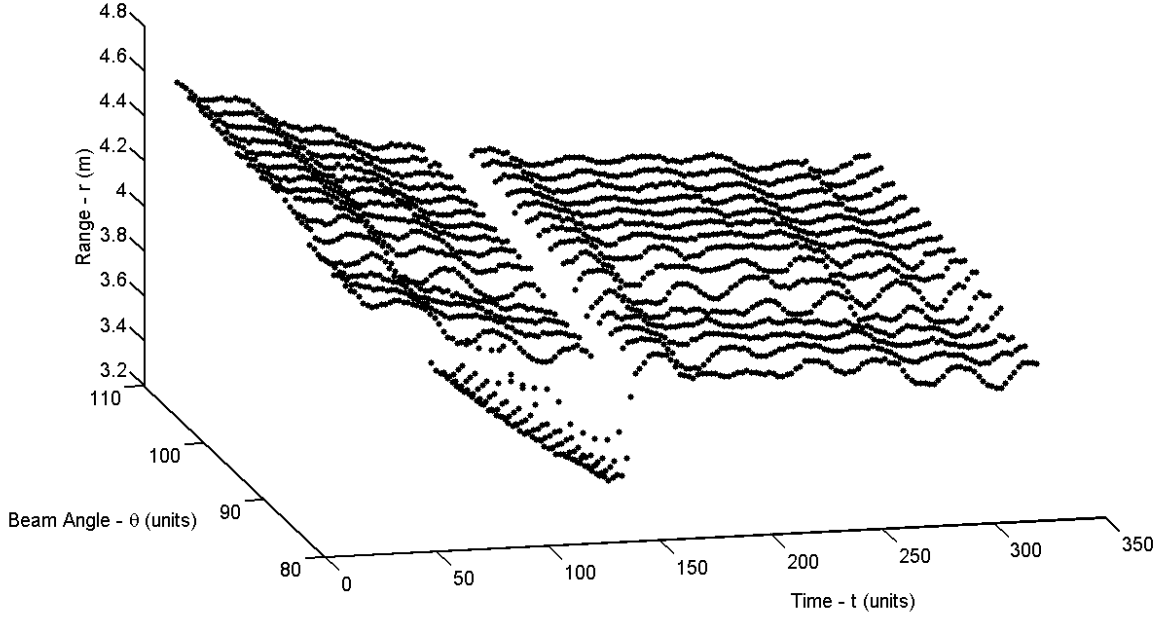


Figure 3.9: Sensor configuration space visualisation of the tree trunk example of Figure 3.7 taken from the New College dataset [97] (as discussed in Chapter 2). Timestamp t and angular position of the laser beam relative to the sensor frame θ are plotted along the horizontal axes. t and θ form the input domain of our regression. Range measurements r form the range of our regression and are presented along the vertical axis in the plot. The apparent waviness of the laser measurements are due to the ground truth trajectory of the sensor which was on a vehicle that rolled and pitched significantly (see Chapter 2 for a description of the data capture vehicle). The general linear decrease in range from left to right indicates that the sensor was moving roughly in a straight line that ended closer to the wall at the right of the example, than at the start. The large dip in range is due to the tree trunk, which was significantly closer to the sensor than the quadrangle wall.

The θ , t spacing of laser measurements across $\mathcal{Q}$ is dependent on the temporal resolution ψ_t and angular resolution ψ_θ , which is constant for our laser scanner ($\frac{1}{75}$ seconds and 0.5° respectively – as given in Table 2.1), hence measurements form a regular grid in θ , t (see the left column of Figure 3.8). Our $\mathcal{Q}$ visualisations are therefore ostensibly

similar to image depth maps. However, image depth maps represent data that is acquired over two spatial dimensions (angles) at a single instance in time with pixels that represent image information over tessellating solid angles. Our $\mathcal{Q}$ visualisation uses one spatial dimension and one temporal dimension (with spatial element due to the movement of the sensor) and each laser range measurement r is formed from averaging range information over a laser beam cone. The spatial relation (in Euclidean space $\mathcal{W}$) between neighbouring laser measurements in $\mathcal{Q}$ (and hence smoothness and mutual dependence properties of laser data in $\mathcal{Q}$) therefore tend to be more complicated and unpredicable than for image range maps.

3.5 Assessing Algorithm Accuracy in Sensor Configuration Space

To assess the algorithms that are presented in this work we compute a success metric that we call *reconstruction error*. To do so we partition the laser data into training data $\mathbf{r}$, $Q = \{\mathbf{q}_i\}_{i=1}^N$ (where N is the number of measurements in the dataset) used to condition our GP regressor, and hold-out measurements $\mathbf{r}_t$, Q_t . Training data is selected in a grid by using one measurement in every two in both dimensions (θ, t) and the hold out set is then randomly sampled from the remaining un-used measurements. The same random sampling is used throughout our work. Error is defined as the ℓ_1 norm (in metres) between the hold out set of laser range measurements $\mathbf{r}_t$ and ranges that are predicted by our algorithm $\mathbf{r}^*$ at the same location in sensor configuration space $\mathcal{Q}$, Q_t .

One would expect that a perfect algorithm would be able to predict the hold out set of measurements with zero reconstruction error. However, the sparse sampling of the workspace means that features such as discontinuities (such as the tree trunk in Figure 3.9) can appear to be at markedly perturbed locations when comparing surfaces that have been generated from training data versus those generated from test data. To gauge the varying effect that this sparse sampling has on reconstruction error we introduce four examples from the New College dataset [97] (see Chapter 2). These test cases are represented by the images in Figures 3.10 and 3.11 which have been taken at roughly the same time that the laser data was captured. We use these test cases throughout Chapters 6 and 7 to allow comparison between algorithms. Surface predictions are generated through linearly interpolating training data $\mathbf{r}$, Q to form laser range predictions $\mathbf{r}^*$ at the hold out positions of sensor configuration space Q_t . The resulting reconstruction error is shown in Figure 3.12 and we use this analysis as a baseline to compare against our algorithm in Chapter 6. Intuitively, laser sparsity makes little difference in simple workspaces such as test case (a) where there are few discontinuities. In test case (b) the errors are marked which is due to many high frequency components in the workspace surface that result from neighbouring laser measurements oscillating between foliage and the walls behind them.

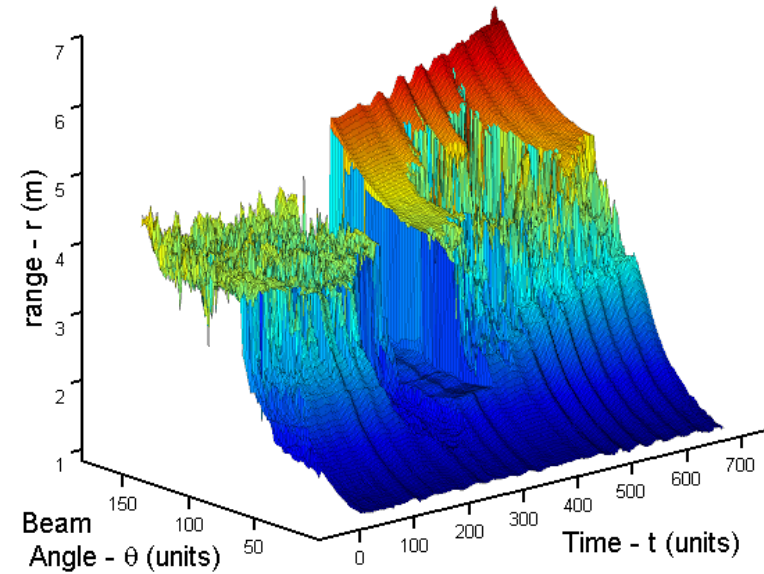
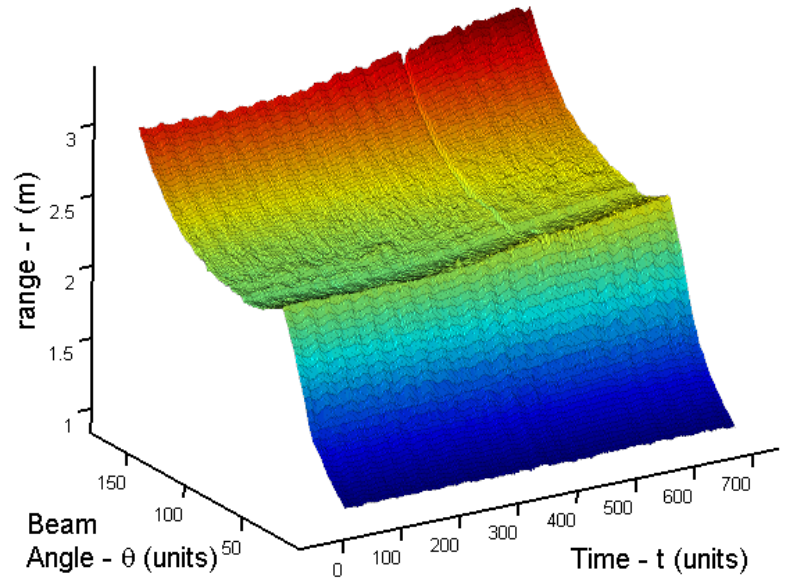


Figure 3.10: Test cases (a) and (b) (row-wise) taken from sections of the New College dataset [97]. These examples are used throughout Chapters 6 and 7. (Left) images were captured at the same time as the laser data that is displayed in sensor configuration space (right). Each surface is constructed through a triangulation of the laser range measurements $\mathbf{r}$ at locations Q . One can clearly discern features in sensor configuration space that align with features in the corresponding images, such as pillars and foliage, despite the dependence between laser ranges and the ground truth trajectory of the sensor. Note, each image is provided for workspace context only and that the field of view of the camera is often different to that of the laser scanner as it traversed its trajectory. This leads to some features appearing in one sensor modality and not the other.

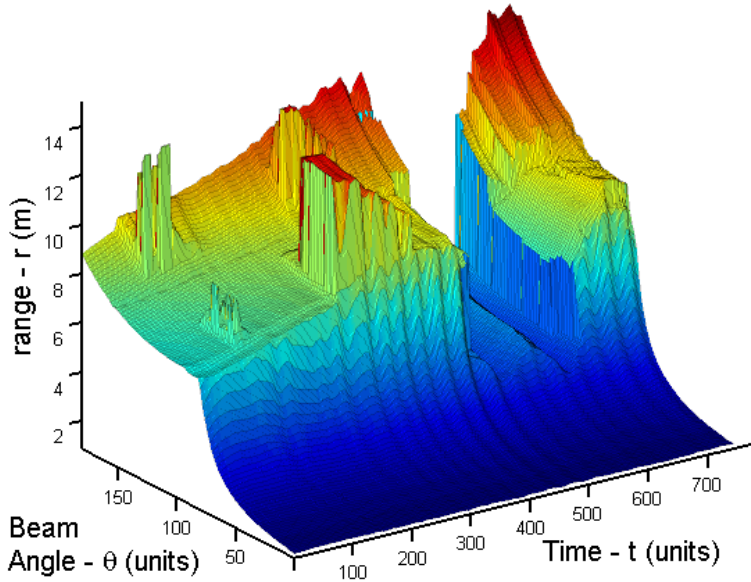
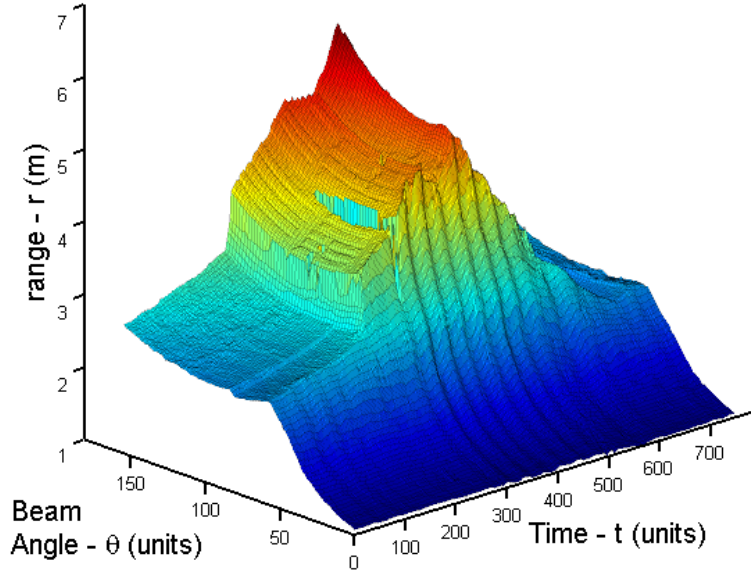


Figure 3.11: Test cases (c) and (d) (row-wise) taken from sections of the New College dataset [97]. These examples are used throughout Chapters 6 and 7. (Left) images were captured at the same time as the laser data that is displayed in sensor configuration space (right). Each surface is constructed through a triangulation of the laser range measurements $\mathbf{r}$ at locations Q . One can clearly discern features in sensor configuration space that align with features in the corresponding images, such as pillars and people, despite the dependence between laser ranges and the ground truth trajectory of the sensor (for example the sensor is turning a corner in test case (c)). Note, each image is provided for workspace context only and that the field of view of the camera is often different to that of the laser scanner as it traversed its trajectory. This leads to some features appearing in one sensor modality and not the other.

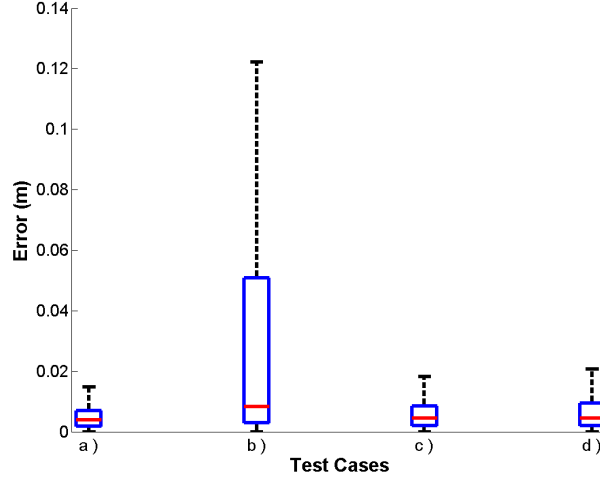


Figure 3.12: Box plots indicating intrinsic reconstruction error as a result of our error assessment approach. There is one box plot for each test case presented in Figures 3.10 and 3.11. Statistics are generated by using linear interpolation to form laser range predictions $\mathbf{r}^*$ at the hold out positions of sensor configuration space Q_t and calculating the error (ℓ_1 norm in $\mathbf{r}$) from $\mathbf{r}_t$. Plots indicate median and interquartile range of the error in $\mathbf{r}$. Whiskers extend a maximum of one and a half times the interquartile range. These results provide intuition about the effect that real data from sparsely sampled workspaces has on our reconstruction error metric, which we will compare to results generated in Chapters 6 and 7. Test case b) has a particularly large median error due to the large amount of foliage in the test case where laser measurements typically alternate between foreground leaves and surrounding walls.

3.6 Summary

In this chapter we have introduced a novel parameterisation of laser point clouds. To inform this parameterisation we first analysed a typical point cloud generated by the mobile robotics platform that we introduced in Chapter 2. We found that although the system produced highly locally consistent point clouds, even the smallest error in the vehicle’s trajectory estimate could lead to a large global inconsistency (as measured by large intra-cluster displacements and angles). To side-step the production of a unified

surface representation under these challenging conditions we forged a simple and novel *local* parameterisation of laser data. This parameterisation allows us to perform surface regression independently to trajectory estimation. Importantly, it also guarantees that our regression is well-formed, and hence compatible with Gaussian Processes. Finally, we investigated the qualities that laser data have in sensor configuration space and suggested apparatus to evaluate surface predictions made therein.

We shall continue our discussion of sensor configuration space in Chapter 5 after first a detailed account of Gaussian Processes in the next chapter. In Chapters 6 and 7 we shall introduce the algorithms that we use to perform our surface regression and analyse their efficacy using the approach outlined above.

Chapter 4

Bayesian Regression Techniques

This chapter provides the theoretical building blocks for the regression techniques that are used throughout this dissertation. Regression analysis is concerned with the formation of predictions of dependent variables given a regression model and independent observations of the variables. We shall in this thesis use it to form predictions of underlying workspace surfaces. These surfaces are based on measurements that are captured from our 3D laser scanning system as it moves through a scene.

We use *Bayesian* inference to form our regression model. That is, rather than using model selection to form point estimates of unknown parameters (following a *frequentist* approach such as Maximum Likelihood) we marginalise these parameters from our models. This helps to avoid the problem of model overfitting (see [13] for a detailed discussion). We do so through the principled application of Bayes rule which additionally provides access to the belief in our predictions in the form of probabilities which we shall make use of throughout our work.

A well known draw back of Bayesian inference stems from finding the marginal of a distribution (through integrating out unwanted random variables) which is often intractable. By adopting the Gaussian as the underlying probability distribution in our frameworks we avoid this problem due to the Gaussian’s consistency property. This property means that the marginals of a multivariate Gaussian distribution are themselves Gaussian, likewise their conditional distributions, which permits us to constrain the space of our computations to that of Gaussians (see Appendix D). This is important practically as the analytical form of multivariate Gaussian distributions are also relatively simple: they are fully specified through second order statistics and for the specific type of *linear Gaussian model* that we consider in our work (see Section 4.2.1) computations require only simple linear algebra.

We also wish our regression models to be more expressive than that is possible through parametric models alone. Instead of relying on a prior over model parameters to induce a distribution over a finite set of parametric basis functions (our model), we also specify a distribution over functions directly through the use of Gaussian Processes. This kernel method again exploits the consistency property of Gaussian distributions to model a potentially infinite set of basis functions in such a way that is tractable.

This chapter proceeds by introducing Bayes rule in Section 4.2. In Section 4.2.1 we derive identities that are essential for both: Bayesian Parametric Regression (Section 4.3), used in Chapter 6; and Gaussian Process Regression (Section 4.4) which is used in Chapters 6 and 7.

4.1 New Nomenclature Introduced in this Chapter

Gaussian Process Regression (GPR)

- $\cdot^*$ - subset of a GP concerning input space $\mathcal{Q}$ that are used for predictions
- $\cdot_\circ$ - vector of infinite length (a function evaluated at all possible locations in its input space)
- $\mathcal{GP}(\mu(\cdot), k(\cdot, \cdot'))$ - Gaussian Process (GP) fully specified by mean function $\mu(\cdot)$ (expectation) and covariance function $k(\cdot, \cdot')$ (variance)
- ξ - regression model hyperparameters
- ζ - regression model parameters - the underlying function $\zeta(\mathbf{q})$
- $p(\zeta|Q)$ - Gaussian Process (GP) prior of the underlying function $\zeta \triangleq \zeta(Q)$ evaluated at the set of training sensor configurations Q
- $p(\mathbf{r}|\zeta, Q)$ - likelihood of range measurements $\mathbf{r}$ given an additive iid Gaussian regression model and underlying function $\zeta(\mathbf{q})$ observed at Q .
- $p(\mathbf{r}|Q)$ - marginal likelihood range measurements $\mathbf{r}$ given an additive iid Gaussian regression model and marginalised underlying function
- $p(r^*|\mathbf{q}^*, \mathcal{D})$ - posterior predictive distribution of range measurement r^* given sensor configuration $\mathbf{q}^*$, data $\mathcal{D}$ and an additive iid Gaussian regression model
- $K(Q, Q)$ - covariance matrix between training data Q (positive semi-definite)
- $K_{\mathbf{r}}$ - augmented covariance matrix $K_{\mathbf{r}} = K(Q, Q) + \sigma_m^2 I$ (positive semi-definite)
- $\mathbf{k}(Q, \mathbf{q}^*)$ - covariance between training points Q and test point (prediction) $\mathbf{q}^*$
- $k(\mathbf{q}^*, \mathbf{q}^*)$ - covariance (variance) of test point (prediction) $\mathbf{q}^*$
- σ_m^2 - measurement variance (sensor accuracy in the regression model)
- σ_p^2 - process noise (scales output space of the covariance function $k(\cdot, \cdot')$)
- l - length scale (scales input space of the covariance function $k(\cdot, \cdot')$)

4.2 Bayes Rule

Bayes rule may be constructed from the basic product rule of probability $p(x, y) = p(y|x)p(x)$ (where x and y are random variables) and symmetry considerations for $p(x, y)$, whose significance was first appreciated by Thomas Bayes in 1763 [10]

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \quad (4.1)$$

where

$$p(x) = \text{marginal probability of } x \text{ (Prior)} \quad (4.2)$$

$$p(y) = \text{marginal probability of } y \text{ (Evidence)} \quad (4.3)$$

$$p(y|x) = \text{conditional probability of } y \text{ given } x \text{ (Likelihood)} \quad (4.4)$$

$$p(x|y) = \text{conditional probability of } x \text{ given } y \text{ (Posterior)} \quad (4.5)$$

Under Bayesian inference a specific meaning is attached to each of the Equations 4.2 through 4.5. Our particular goal is to combine prior knowledge $p(x)$ with additional information in the form of a likelihood $p(y|x)$ (produced through a laser measurement model) to create an updated belief $p(x|y)$, otherwise known as a posterior distribution in x . We use this posterior to form predictions in our regression framework, conditioned on measurements taken by our laser scanner.

4.2.1 Gaussian Variables with Linearly Dependent Means

In our regression frameworks we use a specific *linear Gaussian model* whereby the mean of our likelihood $p(\mathbf{y}|\mathbf{x})$ is a linear function of a multivariate Gaussian distributed random variable $\mathbf{x}$ and the covariance is independent of $\mathbf{x}$

$$p(\mathbf{x}) \triangleq \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \Sigma_{\mathbf{xx}}) \quad (4.6)$$

$$p(\mathbf{y}|\mathbf{x}) \triangleq \mathcal{N}(\mathbf{y}; M\mathbf{x} + \mathbf{c}, \Sigma_{\mathbf{y}|\mathbf{x}}) \quad (4.7)$$

where $p(\mathbf{x})$ is our prior distribution, M is a matrix, $\mathbf{c}$ is a vector and we have adopted the same nomenclature as Appendix D. We may construct the joint distribution $p(\mathbf{z}) = p(\mathbf{x}, \mathbf{y})$ through the basic product rule of probability

$$\begin{aligned} \log(p(\mathbf{x}, \mathbf{y})) &= \log(p(\mathbf{y}|\mathbf{x})) + \log(p(\mathbf{x})) \\ -\frac{1}{2}\mathbf{z}^T \Sigma^{-1} \mathbf{z} &= -\frac{1}{2}(\mathbf{y} - M\mathbf{x} - \mathbf{c})^T \Sigma_{\mathbf{y}|\mathbf{x}}^{-1} (\mathbf{y} - M\mathbf{x} - \mathbf{c}) \dots \\ &\quad -\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}})^T \Sigma_{\mathbf{xx}}^{-1} (\mathbf{x} - \bar{\mathbf{x}}) + \boldsymbol{\alpha} \end{aligned} \quad (4.8)$$

where $\boldsymbol{\alpha}$ are the terms that are independent of $\mathbf{x}$ and $\mathbf{y}$. The result is again a Gaussian distribution in $\mathbf{z}$ of which we may recover the precision matrix $\Lambda = \Sigma^{-1}$ (see Equation D.8) through comparison of second order terms in $\mathbf{z}$

$$-\frac{1}{2}\mathbf{z}^T\Lambda\mathbf{z} = -\frac{1}{2}\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}^T \begin{bmatrix} \Psi^{-1} & -M^T\Sigma_{\mathbf{y}|\mathbf{x}}^{-1} \\ -\Sigma_{\mathbf{y}|\mathbf{x}}^{-1}M & \Sigma_{\mathbf{y}|\mathbf{x}}^{-1} \end{bmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \quad (4.9)$$

where for later convenience we have defined

$$\Lambda_{\mathbf{xx}} = \Psi^{-1} = \Sigma_{\mathbf{xx}}^{-1} + M^T\Sigma_{\mathbf{y}|\mathbf{x}}^{-1}M \quad (4.10)$$

We may translate this precision matrix Λ into the more commonly used covariance matrix Σ through the partitioned matrix formula (see [86]¹)

$$p(\mathbf{x}, \mathbf{y}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}; \begin{bmatrix} \bar{\mathbf{x}} \\ M\bar{\mathbf{x}} + \mathbf{c} \end{bmatrix}, \begin{bmatrix} \Sigma_{\mathbf{xx}} & \Sigma_{\mathbf{xx}}M^T \\ M\Sigma_{\mathbf{xx}} & \Sigma_{\mathbf{y}|\mathbf{x}} + M\Sigma_{\mathbf{xx}}M^T \end{bmatrix}\right) \quad (4.13)$$

where we have similarly found the mean $\bar{\mathbf{z}}$ through comparing linear terms in Equation 4.8. We can extract the evidence $p(\mathbf{y})$ from Equation 4.13 following the identity in Equation D.6 which was created through application of the marginalisation property of Gaussian distributions.

¹We may confirm that these results conform to the more general joint Gaussian distributed variables (discussed in Appendix D) through the use of the Matrix Inversion Lemma [114]. Specifically, when Equations D.14 and D.15 are substituted with appropriate terms from Equation 4.13 they become identical to $\Lambda_{\mathbf{xx}}$ and $\Lambda_{\mathbf{xy}}$ from Equation 4.9

$$\Lambda_{\mathbf{xx}} = \Psi^{-1} = (\Sigma_{\mathbf{xx}} - \Sigma_{\mathbf{xx}}M^T(\Sigma_{\mathbf{y}|\mathbf{x}} + M\Sigma_{\mathbf{xx}}M^T)^{-1}M\Sigma_{\mathbf{xx}})^{-1} \quad (4.11)$$

$$\Lambda_{\mathbf{xy}} = -M^T\Sigma_{\mathbf{y}|\mathbf{x}}^{-1} = -\Psi^{-1}\Sigma_{\mathbf{xx}}M^T(\Sigma_{\mathbf{y}|\mathbf{x}} + M\Sigma_{\mathbf{xx}}M^T)^{-1} \quad (4.12)$$

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}; M\bar{\mathbf{x}} + \mathbf{c}, \Sigma_{\mathbf{y}|\mathbf{x}} + M\Sigma_{\mathbf{xx}}M^T) \quad (4.14)$$

We may then also construct the posterior distribution $p(\mathbf{x}|\mathbf{y})$ through plugging in the relevant block precision matrices from Equation 4.9 into the posterior distribution of Equation D.13.

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}\left(\mathbf{x}; \Psi\left(M^T\Sigma_{\mathbf{y}|\mathbf{x}}^{-1}(\mathbf{y} - \mathbf{c}) + \Sigma_{\mathbf{xx}}^{-1}\bar{\mathbf{x}}\right), \Psi\right) \quad (4.15)$$

4.3 Bayesian Parametric Regression

For Bayesian parametric model fitting over the domain $\mathbf{q} \in \mathcal{Q}$ (where for consistency with Chapter 3 $\mathbf{q}$ is an arbitrary configuration in Sensor Configuration Space $\mathcal{Q}$, although it could equally be any multivariate in space $\mathcal{Q}$) we assume a parametric model for our latent function $f(\mathbf{q})$ of the form

$$f(\mathbf{q}_i) = \mathbf{h}(\mathbf{q}_i)^T \boldsymbol{\beta} \quad (4.16)$$

where

$$p(\boldsymbol{\beta}) = \mathcal{N}(\boldsymbol{\beta}; \bar{\mathbf{b}}, B) \quad (4.17)$$

$$\boldsymbol{\beta} = [\beta_1, \beta_2 \dots \beta_e]^T \quad (4.18)$$

$$\bar{\mathbf{b}} = \mathbb{E}[\boldsymbol{\beta}] \quad (4.19)$$

$$B = \mathbb{V}[\boldsymbol{\beta}] = \mathbb{E}[(\boldsymbol{\beta} - \bar{\mathbf{b}})^T (\boldsymbol{\beta} - \bar{\mathbf{b}})] \quad (4.20)$$

We have used the index i to select single configurations $\mathbf{q}_i$ and the corresponding value of $f(\mathbf{q})$ evaluated at $\mathbf{q}_i$, and we use the same nomenclature for Expectation $\mathbb{E}$ and Variance $\mathbb{V}$ as Appendix D. $\mathbf{h}$ is a vector of basis functions of length e (for example a polynomial model could employ $\mathbf{h}(x) = [1, x, x^2 \dots]^T$ and a sinusoidal model $\mathbf{h}(x) = [1, \sin(x), \sin(2x) \dots]^T$). We have assigned a prior distribution $p(\boldsymbol{\beta})$ over basis function weightings $\boldsymbol{\beta}$ which induces a distribution over $f(\mathbf{q}_i)$. Our set of model parameters are therefore $\boldsymbol{\zeta} = \{\bar{\mathbf{b}}, B, \mathbf{h}\}$.

For Bayesian regression we assume additive Gaussian noise $\epsilon \sim \mathcal{N}(\epsilon; 0, \sigma_m^2)$ to account for measurement inaccuracies in our laser sensor. Our linear regression model is therefore

$$\begin{aligned} r_i &= \mathbf{h}(\mathbf{q}_i)^T \boldsymbol{\beta} + \epsilon_i \\ p(r_i | \mathbf{q}_i, \boldsymbol{\beta}) &= \mathcal{N}(r_i; \mathbf{h}(\mathbf{q}_i)^T \boldsymbol{\beta}, \sigma_m^2) \end{aligned} \quad (4.21)$$

where each measurement is r_i corresponding to $\mathbf{q}_i$. Assuming that measurements

$\mathbf{r} = [r_1, r_2 \dots r_N]^T$ have been drawn independently from Equation 4.21 at locations $Q = \{\mathbf{q}_i\}_{i=1}^N$ the likelihood $p(\mathbf{r}|\boldsymbol{\beta})$ is therefore

$$\begin{aligned}\mathbf{r} &= H^T \boldsymbol{\beta} + \boldsymbol{\epsilon} \\ p(\mathbf{r}|Q, \boldsymbol{\beta}) &= \prod_{i=1}^N \mathcal{N}(r_i; \mathbf{h}(\mathbf{q}_i)^T \boldsymbol{\beta}, \sigma_m^2) \\ &= \mathcal{N}(\mathbf{r}; H^T \boldsymbol{\beta}, \Omega)\end{aligned}\tag{4.22}$$

where

$$H = [\mathbf{h}(\mathbf{q}_1), \mathbf{h}(\mathbf{q}_2) \dots \mathbf{h}(\mathbf{q}_N)]\tag{4.23}$$

$$\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2 \dots \epsilon_N]^T \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \Omega)\tag{4.24}$$

$$\Omega = \sigma_m^2 I\tag{4.25}$$

We may update our prior belief over model weightings (Equation 4.17) with information from our measurements through our likelihood (Equation 4.22) as per Bayes rule (Equation 4.2). As our likelihood has a linear dependency on the prior we may use the posterior distribution in Equation 4.15 to arrive at the posterior distribution of our model given training data $\mathcal{D} = \{(\mathbf{q}_i, r_i)\}_{i=1}^N$

$$\begin{aligned}
p(\boldsymbol{\beta}|\mathcal{D}) &= \frac{p(\mathbf{r}|\boldsymbol{\beta}, Q)p(\boldsymbol{\beta}, Q)}{p(\mathcal{D})} \\
&= \mathcal{N}(\boldsymbol{\beta}; \bar{\boldsymbol{\beta}}, A^{-1})
\end{aligned} \tag{4.26}$$

where

$$A = B^{-1} + H\Omega^{-1}H^T \tag{4.27}$$

$$\bar{\boldsymbol{\beta}} = A^{-1}(H\Omega^{-1}\mathbf{r} + B^{-1}\bar{\mathbf{b}}) \tag{4.28}$$

For Bayesian inference we wish to form predictions r^* (at a test point $\mathbf{q}^*$ in our domain $\mathcal{Q}$ and conditioned on our training data $\mathcal{D}$) that are independent of our parameter weightings $\boldsymbol{\beta}$. We achieve through marginalisation of $\boldsymbol{\beta}$ using our regression model $p(r^*|\mathbf{q}^*, \boldsymbol{\beta})$ (Equation 4.21) for our predictions of r^* .

$$\begin{aligned}
p(r^*|\mathbf{q}^*, \mathcal{D}) &= \int p(r^*, \boldsymbol{\beta}|\mathbf{q}^*, \mathcal{D}) d\boldsymbol{\beta} = \int p(r^*|\mathbf{q}^*, \boldsymbol{\beta}, \mathcal{D})p(\boldsymbol{\beta}|\mathbf{q}^*, \mathcal{D}) d\boldsymbol{\beta} \\
&= \int p(r^*|\mathbf{q}^*, \boldsymbol{\beta})p(\boldsymbol{\beta}|\mathcal{D}) d\boldsymbol{\beta}
\end{aligned} \tag{4.29}$$

Noting that $p(r^*|\mathbf{q}^*, \boldsymbol{\beta})$ has linear dependence on the posterior $p(\boldsymbol{\beta}|\mathcal{D})$ we may use the marginal distribution of Equation 4.14 to arrive at our posterior predictive distribution

$$p(r^*|\mathbf{q}^*, \mathcal{D}) = \mathcal{N}(r^*; \mathbf{h}(\mathbf{q}^*)^T \bar{\boldsymbol{\beta}}, \sigma_m^2 + \mathbf{h}(\mathbf{q}^*)^T A^{-1} \mathbf{h}(\mathbf{q}^*)) \tag{4.30}$$

4.4 Gaussian Process Regression (GPR)

In the previous section we explicitly defined basis functions $\mathbf{h}$ and a prior distribution over their associated weightings $p(\boldsymbol{\beta}) = \mathcal{N}(\boldsymbol{\beta}; \bar{\mathbf{b}}, B)$ to induce a distribution over our regression model. In this section we dispense with these parameters and instead take a *function space* viewpoint by specifying a distribution over functions (our model) directly through the use of Gaussian Processes (GPs). This permits us, with an appropriately chosen GP, not only to model certain parametric models, but also non-parametric models through use of the *kernel trick* (alternatively known as *kernel substitution* - see [88] or [13]). Assuming, for ease of demonstration, a zero mean prior ($\bar{\mathbf{b}} = 0$ in Equation 4.17) the mean of the parametric predictive distribution (Equation 4.30) may be re-written as a linear combination of the training measurements $\mathbf{r}$

$$\mathbb{E}[r^* | \mathbf{q}^*, \mathcal{D}] = \mathbf{h}(\mathbf{q}^*)^T A^{-1} H \Omega^{-1} \mathbf{r} \quad (4.31)$$

$$= \sum_{i=1}^N \sigma_m^{-2} \mathbf{h}(\mathbf{q}^*)^T A^{-1} \mathbf{h}(\mathbf{q}_i) r_i \quad (4.32)$$

$$= \sum_{i=1}^N \delta(\mathbf{q}^*, \mathbf{q}_i) r_i \quad (4.33)$$

By noting that the inner product in Equation 4.32 we may write the compact kernel notation where $\delta(\mathbf{q}^*, \mathbf{q}_i)$ is a an *equivalent kernel*, or *smoother matrix* (see [13] for further discussion). As we shall confirm in Equation 4.49, GPs allow us to specify this kernel while allowing a potentially infinite number of basis functions, thereby providing us with a powerful regression technique.

A Gaussian Process consists of a collection of random variables, a finite subset of which have a joint Gaussian distribution. It is fully specified by a mean $\mu(\mathbf{q})$ and a covariance $k(\mathbf{q}, \mathbf{q}')$ function

$$\zeta(\mathbf{q}) \sim \mathcal{GP}(\mu(\mathbf{q}), k(\mathbf{q}, \mathbf{q}')) \quad (4.34)$$

where $k(\mathbf{q}, \mathbf{q}')$ is the covariance between two corresponding points $\mathbf{q}$ and $\mathbf{q}'$ in $\mathcal{Q}$ (we shall discuss suitable mean and covariance functions in Section 4.4.1). A GP prior provides a compatible substitute² for the distribution over our parametric regression model $f(\mathbf{q})$ (as introduced in Equation 4.16). We adopt the notation $\zeta(\mathbf{q})$ to indicate that the parameters in this section are embodied in the non-parametric model $\zeta(\mathbf{q})$ itself - that is, one may consider the model as being parametrised by an infinitely long vector of outputs $\zeta_{\circ}$ (where $\zeta_{\circ} \triangleq \zeta(\mathbf{q}_{\circ})$) with inputs $\mathbf{q}_{\circ}$ spanning the input space $\mathcal{Q}$.³

The consistency property of Gaussian Distributions (see Appendix D) allows us to consider the values of the uncountably infinite space of functions at only the discrete set of training values $\zeta \triangleq \zeta(Q)$ (where $Q = \{\mathbf{q}_i\}_{i=1}^N$). We may hence collapse the joint distribution of Equation 4.34 to

$$p(\zeta|Q) = \mathcal{N}(\zeta; \mu(Q), K) \quad (4.35)$$

where

²Specifically, for $\beta \sim \mathcal{N}(\beta; \mathbf{0}, B)$, $f(\mathbf{q})$ and $f(\mathbf{q}')$ (of Section 4.3) are joint Gaussian distributed

$$\begin{aligned} \mathbb{E}[f(\mathbf{q})] &= \mathbf{h}(\mathbf{q})^T \mathbb{E}[\beta] = \mathbf{0} \\ \mathbb{E}[f(\mathbf{q})f(\mathbf{q}')] &= \mathbf{h}(\mathbf{q})^T \mathbb{E}[\beta\beta^T] \mathbf{h}(\mathbf{q}') = \mathbf{h}(\mathbf{q})^T B \mathbf{h}(\mathbf{q}') \end{aligned}$$

³Although the $\circ$ notation is not used in earnest until Chapter 7 we introduce it here for consistency.

$$\begin{aligned}
K &= K(Q, Q) \text{ (size } N \times N \text{) covariance matrix} \\
&\text{between all training points } Q
\end{aligned} \tag{4.36}$$

$$\boldsymbol{\mu}(Q) = [\mu(\mathbf{q}_1), \mu(\mathbf{q}_2) \dots \mu(\mathbf{q}_N)]^T \tag{4.37}$$

and for future convenience we also define

$$\begin{aligned}
\mathbf{k}(Q, \mathbf{q}^*) &= \text{(size } N \times 1 \text{) covariance between training} \\
&\text{points } Q \text{ and test point (prediction) } \mathbf{q}^*
\end{aligned} \tag{4.38}$$

$$\begin{aligned}
k(\mathbf{q}^*, \mathbf{q}^*) &= \text{(size } 1 \times 1 \text{) covariance (variance) of} \\
&\text{test point (prediction) } \mathbf{q}^*
\end{aligned} \tag{4.39}$$

Each element of the covariance terms (Equations 4.36, 4.38 and 4.39) are constructed from the covariance function $k(\mathbf{q}, \mathbf{q}')$ of Equation 4.34 and our training values Q where, for example $K_{ij} = k(\mathbf{q}_i, \mathbf{q}_j)$.⁴ We adopt the same additive iid Gaussian noise regression model as Section 4.3 to form the likelihood $p(\mathbf{r}|\boldsymbol{\zeta}, Q)$

$$\begin{aligned}
\mathbf{r} &= \boldsymbol{\zeta} + \boldsymbol{\epsilon} \\
p(\mathbf{r}|\boldsymbol{\zeta}, Q) &= \mathcal{N}(\mathbf{r}; \boldsymbol{\zeta}, \sigma_m^2 I)
\end{aligned} \tag{4.40}$$

⁴The symmetry in K permits us to avoid explicit computation of every element.

We may now marginalise our model parameters $\boldsymbol{\zeta}$ directly without recourse to explicitly forming a posterior distribution (which was required in Section 4.3 - see Equations 4.26 and 4.29).

$$p(\mathbf{r}|Q) = \int p(\mathbf{r}, \boldsymbol{\zeta}|Q) d\boldsymbol{\zeta} = \int p(\mathbf{r}|\boldsymbol{\zeta}, Q)p(\boldsymbol{\zeta}|Q) d\boldsymbol{\zeta} \quad (4.41)$$

As the likelihood $p(\mathbf{r}|\boldsymbol{\zeta}, Q)$ has a mean that has trivial linear dependence on the prior $p(\boldsymbol{\zeta}|Q)$ (the identity) we may use the linear distribution of Equation 4.14 to arrive at the marginal likelihood $p(\mathbf{r}|Q)$

$$p(\mathbf{r}|Q) = \mathcal{N}(\mathbf{r}; \boldsymbol{\mu}(Q), K_{\mathbf{r}}) \quad (4.42)$$

$$\begin{aligned} \log(p(\mathbf{r}|Q)) &= -\frac{1}{2} \left[(\mathbf{r} - \boldsymbol{\mu}(Q))^T K_{\mathbf{r}}^{-1} (\mathbf{r} - \boldsymbol{\mu}(Q)) \dots \right. \\ &\quad \left. + \log |K_{\mathbf{r}}| + N \log(2\pi) \right] \end{aligned} \quad (4.43)$$

where

$$K_{\mathbf{r}} = K + \sigma_m^2 I \quad (4.44)$$

The marginal likelihood is an important distribution that encodes how well our measurements fit our model. We shall make extensive use of $p(\mathbf{r}|Q)$ for model selection and averaging in Chapters 5 and 7 respectively.

To form predictions conditioned on our training data $\mathcal{D} = \{(\mathbf{q}_i, r_i)\}_{i=1}^N$ we first form the joint distribution with the prediction r^* at location $\mathbf{q}^*$. By assuming the same

model for our predictions as for our training data (which led to the likelihood function in Equation 4.40) and by using the consistency property of Gaussian Distributions (see Appendix D) we may write

$$p(\mathbf{r}, r^* | \mathbf{q}^*, Q) = \mathcal{N} \left(\begin{bmatrix} \mathbf{r} \\ r^* \end{bmatrix} ; \begin{bmatrix} \boldsymbol{\mu}(Q) \\ \mu(\mathbf{q}^*) \end{bmatrix}, \begin{bmatrix} K_{\mathbf{r}} & \mathbf{k}(Q, \mathbf{q}^*) \\ \mathbf{k}(Q, \mathbf{q}^*)^T & k(\mathbf{q}^*, \mathbf{q}^*) + \sigma_m^2 \end{bmatrix} \right) \quad (4.45)$$

From this joint distribution we can arrive at the posterior predictive distribution of our regression model through the posterior distribution in Equation D.16

$$p(r^* | \mathbf{q}^*, \mathcal{D}) = \mathcal{N}(r^*; \mathbb{E}[r^* | \mathbf{q}^*, \mathcal{D}], \mathbb{V}[r^* | \mathbf{q}^*, \mathcal{D}]) \quad (4.46)$$

where

$$\begin{aligned} \mathbb{E}[r^* | \mathbf{q}^*, \mathcal{D}] &= \mu(\mathbf{q}^*) + \mathbf{k}(Q, \mathbf{q}^*)^T K_{\mathbf{r}}^{-1} (\mathbf{r} - \boldsymbol{\mu}(Q)) \\ &= \mathbb{E}[r^* | \mathbf{q}^*, \mathcal{D}]_{\mu(\mathbf{q})=0} + \mu(\mathbf{q}^*) - \mathbf{k}(Q, \mathbf{q}^*)^T K_{\mathbf{r}}^{-1} \boldsymbol{\mu}(Q) \end{aligned} \quad (4.47)$$

$$\begin{aligned} \mathbb{V}[r^* | \mathbf{q}^*, \mathcal{D}] &= k(\mathbf{q}^*, \mathbf{q}^*) + \sigma_m^2 - \mathbf{k}(Q, \mathbf{q}^*)^T K_{\mathbf{r}}^{-1} \mathbf{k}(Q, \mathbf{q}^*) \\ &= \mathbb{V}[r^* | \mathbf{q}^*, \mathcal{D}]_{\mu(\mathbf{q})=0} \end{aligned} \quad (4.48)$$

$\mathbb{E}[r^* | \mathbf{q}^*, \mathcal{D}]_{\mu(\mathbf{q})=0}$ and $\mathbb{V}[r^* | \mathbf{q}^*, \mathcal{D}]_{\mu(\mathbf{q})=0}$ are the mean and variance of the posterior predictive distribution assuming a zero mean function ($\mu(\mathbf{q}) = 0$). By noting that

$\mathbb{E}[r^*|\mathbf{q}^*, \mathcal{D}]_{\mu(\mathbf{q})=0}$ is simply a linear combination of training measurements $\mathbf{r}$ we can confirm that GPs can be used to encode an equivalent kernel in a compact manner, as stipulated at the beginning of this section:

$$\begin{aligned}\mathbb{E}[r^*|\mathbf{q}^*, \mathcal{D}]_{\mu(\mathbf{q})=0} &= \mathbf{k}(Q, \mathbf{q}^*)^T K_{\mathbf{r}}^{-1} \mathbf{r} \\ &= \sum_{i=1}^N \gamma_i k(\mathbf{q}_i, \mathbf{q}^*)\end{aligned}\tag{4.49}$$

where

$$\boldsymbol{\gamma} = K_{\mathbf{r}}^{-1} \mathbf{r}\tag{4.50}$$

4.4.1 Hyperparameters

As shown, GPs can be used to perform non-parametric regression as they enable the tractable marginalisation of parametric model parameters $\boldsymbol{\zeta}$. A GP may be defined by specifying the family of permissible underlying functions $\zeta(\mathbf{q})$ through the GP covariance function $k(\mathbf{q}, \mathbf{q}')$ (see Equation 4.34). Any expectation of the underlying function may also be specified through the GP mean function $\mu(\mathbf{q})$ (also found in Equation 4.34). The model parameters which specify these functions are termed hyperparameters $\boldsymbol{\xi}$ to distinguish them from the parametric model parameters $\boldsymbol{\zeta}$ of Section 4.3. Hyperparameters $\boldsymbol{\xi}$ specify only the general form of the underlying function in contrast to parameters $\boldsymbol{\zeta}$ that directly specify the function. Non-parametric regression tends to be more robust than parametric regression under the presence of non-optimally set model parameters.

We observe this robustness when we compare a non-parametric regression model with fixed hyperparameter values against a more expressive non-parametric models with hyperparameter values that vary in response to local workspace conditions in Chapter 7.

4.4.1.1 Mean Function

In the absence of prior knowledge about the mean of $\mathbf{r}$ and by symmetry arguments, the mean function is often assumed to be zero $\mu(\mathbf{q}) = 0$. However, it can sometimes be advantageous to incorporate stronger assumptions of the underlying workspace function that we wish to model. In Section 6.5 we discuss a particular example of a parametric mean function that is constructed from a handful of explicit basis functions and a prior over their weightings that is infinitely vague.

4.4.1.2 Covariance Function

An arbitrary kernel function $\delta(\mathbf{q}, \mathbf{q}')$ will not in general be a valid GP covariance function $k(\mathbf{q}, \mathbf{q}')$. A necessary and sufficient condition is that the *gram matrix* Δ (constructed from the kernel function, where $\Delta_{ij} = \delta(\mathbf{q}_i, \mathbf{q}_j)$) is positive semidefinite (PSD) and in which case the matrix is a valid GP covariance matrix K (as used in Section 4.4)⁵. Despite this limitation there are numerous covariance functions (see [88]) that each encode different assumptions of the underlying function that we wish to model. Like general kernel functions, valid covariance functions may also be combined into more

⁵Note: this assumes a non-zero measurement noise in our regression model $\sigma_m^2 > 0$. A symmetric matrix K is PSD if $\mathbf{x}^T K \mathbf{x} \geq 0$ for all non-zero vectors $\mathbf{x}$. This is true if, for example, all the eigenvalues of K are non-negative.

complex ones (see Chapter 7 for an example of where this may be expedient for a particular task at hand and [13] for a list of permissible operators). We shall now discuss only a subset of the most popular covariance functions that we use throughout this dissertation.

A common assumption is that the covariance function is a function of $\mathbf{q} - \mathbf{q}'$. These covariance functions are invariant to translations in the input space $\mathcal{Q}$ and are therefore termed *stationary*. The prototypical example of a stationary covariance function is the Squared Exponential $k(\mathbf{q}, \mathbf{q}')_{SE}$

$$k(\mathbf{q}, \mathbf{q}')_{SE} = \exp\left(-\frac{d^2}{2}\right) \quad (4.51)$$

where

$$d(\mathbf{q}, \mathbf{q}') = \|\mathbf{q} - \mathbf{q}'\|_{\mathcal{Q}} = \sqrt{(\mathbf{q} - \mathbf{q}')^T(\mathbf{q} - \mathbf{q}')} \quad (4.52)$$

This covariance function is sometimes termed a *Radial Basis Function* or is said to be *isotropic* (or *homogeneous* according to [13]) as it is a function of only the magnitude of the difference of locations in input space $\|\mathbf{q} - \mathbf{q}'\|_{\mathcal{Q}}$ (where we have used the Euclidean distance as defined Equation 4.52). It is therefore invariant to all rigid motions in the input space. In practice we introduce hyperparameters $\boldsymbol{\xi} = \{l, \sigma_p\}$ to control the otherwise arbitrary scaling of both the input and output spaces (l and σ_p respectively).

$$k(\mathbf{q}, \mathbf{q}')_{SE} = \sigma_p^2 \exp\left(-\frac{d^2}{2l^2}\right) \quad (4.53)$$

These hyperparameters have intuitive interpretations in relation to the understand-

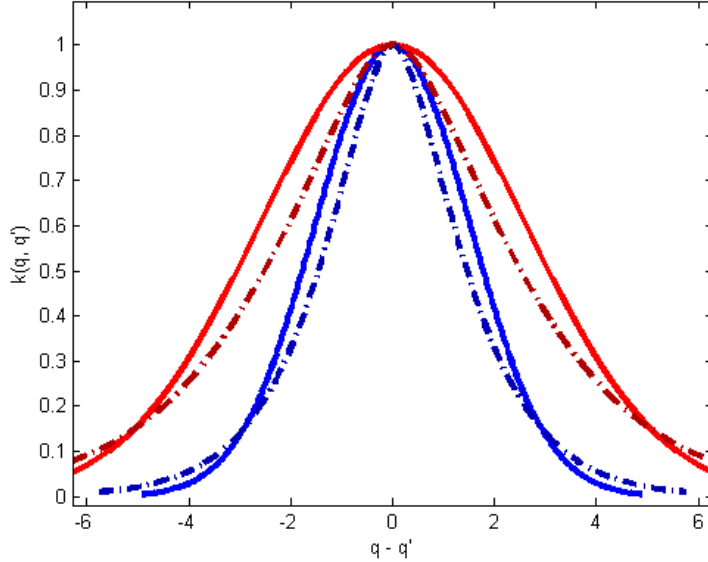


Figure 4.1: Visualisation of the Matérn family of covariance functions $k_{\text{Matérn}}$ with two different smoothness ν and lengthscale l hyperparameters. For simplicity we assume $\mathbf{q} \in \mathcal{R}^1$ and have fixed process variance $\sigma_p^2 = 1$. Blue lines correspond to $l = 1.5$ and red $l = 2.6$, while solid lines correspond to k_{SE} ($\nu \rightarrow \infty$) and dashed $\nu = \frac{3}{2}$. Plots become more peaked (and therefore less smooth) as either l or ν decrease.

ding of the data $\mathcal{D}$. The larger the process variance σ_p the larger the expected variability in the output space, and the larger the length scale l the more smoothly (less rapidly) we expect the underlying function to vary with respect to the input space. By smoothness we are loosely concerned with the distance between $\mathbf{q}$ and $\mathbf{q}'$ in input space that the associated underlying function values become uncorrelated and thus unable to influence each other⁶. In Figure 4.1 we see that for k_{SE} that this distance (and therefore “smoothness”) increases as l increases.

⁶Technically, k_{SE} is said to be *smooth* due to being infinitely differentiable (and hence results in an underlying function that is infinitely mean square differentiable - see [88]). Therefore k_{SE} has continuity of all derivatives around $q - q' = 0$, unlike $k_{\text{Matérn}}$ (Equation 4.54) which results in an underlying function that is only α -times mean square differentiable for $\nu > \alpha$ - see Figure 4.1.

Stein [103] argues that the strong smoothness assumption that is enforced by k_{SE} is inappropriate for many physical processes. Instead he suggested the Matérn family of covariance function $k_{Matérn}$ that has variable smoothness, which is controlled by hyperparameter $\nu > 0$

$$k(\mathbf{q}, \mathbf{q}')_{Matérn} = \sigma_p^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{d\sqrt{2\nu}}{l} \right)^\nu K_\nu \left(\frac{d\sqrt{2\nu}}{l} \right) \quad (4.54)$$

where $\Gamma(z) = \int_0^\infty t^{z-1} \exp(-t) dt$ is the Gamma function, K is the modified Bessel function of the second kind (see [2]) and l , σ_p^2 and d are as above. Fortunately Equation 4.54 simplifies to the product of an exponential and a polynomial for half-integer ν

$$k(\mathbf{q}, \mathbf{q}')_{Matérn \nu=\frac{1}{2}} = \sigma_p^2 \exp \left(-\frac{d\sqrt{3}}{l} \right) = k(\mathbf{q}, \mathbf{q}')_E \quad (4.55)$$

$$k(\mathbf{q}, \mathbf{q}')_{Matérn \nu=\frac{3}{2}} = \sigma_p^2 \left(1 + \frac{d\sqrt{3}}{l} \right) \exp \left(-\frac{d\sqrt{3}}{l} \right) \quad (4.56)$$

$$k(\mathbf{q}, \mathbf{q}')_{Matérn \nu=\frac{5}{2}} = \sigma_p^2 \left(1 + \frac{d\sqrt{5}}{l} + \frac{5d^2}{3l^2} \right) \exp \left(-\frac{d\sqrt{5}}{l} \right) \quad (4.57)$$

$$k(\mathbf{q}, \mathbf{q}')_{Matérn \nu \rightarrow \infty} = \sigma_p^2 \exp \left(-\frac{d^2}{2l^2} \right) = k(\mathbf{q}, \mathbf{q}')_{SE} \quad (4.58)$$

Smoothness increases with ν from the very rough Exponential covariance function $k(\mathbf{q}, \mathbf{q}')_E$ when $\nu = \frac{1}{2}$ to the very smooth Squared Exponential covariance function $k(\mathbf{q}, \mathbf{q}')_{SE}$ as $\nu \rightarrow \infty$. As noted in [88] it is probably very hard to distinguish between $\nu \geq \frac{7}{2}$ and $\nu \rightarrow \infty$ from a finite number of noisy training data and the underlying

function becomes very rough below $\nu = \frac{1}{2}$.⁷ We plot $k(\mathbf{q}, \mathbf{q}')_{\text{Matérn } \nu=\frac{3}{2}}$ and $k(\mathbf{q}, \mathbf{q}')_{SE}$ in Figure 4.1 to allow comparison of kernels.

4.4.2 Practical Considerations for Covariance Matrices

Theoretically a large set of noise free measurements $\mathbf{r}$ that are finely separated in input space $\mathcal{Q}$ would be ideal for the production of highly accurate predictions. Each prediction would be able to draw upon a large number of support measurements which would all inform the local surface conditions of the underlying functions. Not only is this often infeasible for many practical processes, but this also requires the inversion of a very large covariance matrix $K_{\mathbf{r}}$. Standard matrix inversion has complexity $\mathcal{O}(N^3)$ (where N is the number of measurements) and hence it would take most modern desktop computers a non-inconsequential amount of time to make predictions for systems with more than a few thousand measurements. We return to this important consideration and techniques to create efficient implementations throughout Chapters 6 and 7.

Even more damaging to our predictions however is that the expectations in Equations 4.47 and 4.48 are sensitive to poor conditioning of $K_{\mathbf{r}}$ (given by a large ratio between the largest and smallest eigenvalue of $K_{\mathbf{r}}$). As discussed in the previous section, any smoothness assumption of the underlying function implies a correlation between nearby measurements. As separation between neighbouring measurements tends towards zero, $K_{\mathbf{r}}$ tends towards singularity (the associated rows and columns become very similar), and hence it becomes very difficult to invert. Although this can be alleviated through the choice of a less smooth covariance function (thereby decreasing

⁷For example, the $k(\mathbf{q}, \mathbf{q}')_E$ is used in the Ornstein-Uhlenbeck process which was introduced to model a particle undergoing Brownian motion [109].

the ratio of length scale to measurement density) and by inflating the measurement noise σ_m^2 (thereby increasing the diagonal elements of $K_{\mathbf{r}}$ by an amount termed *jitter* [66]) this artificially restricts the expressive capacity of the regression framework. Likewise, the Moore-Penrose pseudoinverse could be used to approximate the inverse of $K_{\mathbf{r}}$ (whereby singular values are zeroed during the Singular Value Decomposition) but this is slow to compute. For covariance functions such as those in Equations 4.55, 4.56, 4.57 and 4.58 derivative observations can also be used in lieu of direct measurements to decrease conditioning. This is because the covariance between a measurement and the partial derivative of a neighbouring measurement is small compared to the covariance between the measurements directly. Noting that differentiation is a linear operator these observations may be readily incorporated in the regression framework by using the results of our linear Gaussian Model considerations in Section 4.2.1. Where derivative observations are not directly observable (as is the case for our work) they have been successfully approximated by taking the gradient at the midpoint between neighbouring observations (see [77]).

We avoid the consequences of poor conditioning and any approximations through the use of Cholesky Decomposition to implicitly invert $K_{\mathbf{r}}$, as is common in GP Literature (see [88]). Since $K_{\mathbf{r}}$ is positive definite we may construct it from its upper triangular Cholesky Factor $C_{\mathbf{r}}$

$$K_{\mathbf{r}} = C_{\mathbf{r}}^T C_{\mathbf{r}} \tag{4.59}$$

This Cholesky factor may be used instead of $K_{\mathbf{r}}$ in our predictions where, for example the mean of our predictions $\mathbb{E}[r^*|\mathbf{q}^*, \mathcal{D}]$ (Equation 4.47) may be computed by

$$\begin{aligned}
\mathbb{E}[r^*|\mathbf{q}^*, \mathcal{D}] &= \mu(\mathbf{q}^*) + \mathbf{k}(Q, \mathbf{q}^*)^T (C_{\mathbf{r}} \setminus (C_{\mathbf{r}}^T \setminus (\mathbf{r} - \boldsymbol{\mu}(Q)))) \\
&= \mu(\mathbf{q}^*) + (C_{\mathbf{r}}^T \setminus \mathbf{k}(Q, \mathbf{q}^*))^T (C_{\mathbf{r}}^T \setminus (\mathbf{r} - \boldsymbol{\mu}(Q))) \quad (4.60)
\end{aligned}$$

We have used $\setminus$ to indicate where forward substitution is used to solve for example $x = Y \setminus z$ in the equation $Yx = z$, where Y is lower triangular of dimension $\alpha \times \alpha$. Forward substitution exploits the triangular structure of Y by solving in order from the top to the bottom of x using the results from each successive row to solve following rows. This operation is $\mathcal{O}(\alpha^2)$ and is more stable under poor conditioning of Y than performing explicit matrix inversions. The second line of Equation 4.60 also indicates how calculations can be ordered so that they can be re-used across multiple equations. For example the solution to $C_{\mathbf{r}}^T \setminus \mathbf{k}(Q, \mathbf{q}^*)$ may be used in the calculation of the predicted variance in Equation 4.48. Overall the complexity of this approach is $\mathcal{O}(N^3)$, dominated by the Cholesky decomposition of Equation 4.59.

Calculation of $C_{\mathbf{r}}$ also provides us efficient access to the determinant of $K_{\mathbf{r}}$ as required in the calculation of the log Marginal Likelihood (Equation 4.43) as used in Chapter 7:

$$\frac{1}{2} \log |K_{\mathbf{r}}| = \sum_{i=1}^N \log(C_{\mathbf{r}_{i,i}}) \quad (4.61)$$

Chapter 5

Gaussian Process Regression in Sensor Configuration Space

This application of GPs to our sensor configuration space parameterisation was first published in ‘Efficient Non-Parametric Surface Representations Using Active Sampling for Push Broom Laser Data’ in the Proceedings of Robotics: Science and Systems, 2010.

This chapter states the assumptions that we make while applying Gaussian Process Regression in sensor configuration space. We also briefly explore GP covariance functions that are appropriate to the regression task at hand and then discuss alternative strategies that are capable of selecting covariance functions automatically.

5.1 Model Assumptions for GPs in Sensor Configuration Space

We assume that each laser measurement r in sensor configuration space $\mathcal{Q}$ is joint Gaussian distributed with covariances derived from a governing covariance function. Together, with an appropriate mean function, this covariance function defines a GP that we use as the main machinery for the regression of range r over $\mathcal{Q}$ (see Section 4.4). We make no attempt to incorporate any knowledge of the spatial relation of neighbouring measurements in Euclidean Space $\mathcal{W}$ (discussed in Section 3.4) or the laser capture method (discussed in Section 2.4.1) in the construction of our covariance function. Instead, we assume a model that uses the natural spacing of the laser measurements in sensor configuration space and a simple stationary covariance function with hyperparameters that reflect various properties of the laser data. Many other covariance functions, including non-stationary covariance functions [52], could be equally adapted and substituted into its place. In Chapter 7 we explore one such non-stationary version which allows the GP hyperparameters to vary across the input domain according to the laser data. As future work (Section 8.1.2) we discuss how more complex measurement models could also be incorporated into our regression framework.

5.2 Simple Stationary Covariance Function for Push Broom Laser Systems

To simplify our implementation we exploit the discrete spacing that arises because the resolutions of our laser scanner are fixed (see Section 2.1 and 3.4). Specifically, we scale both dimensions $\{\theta, t\}$ of $\mathcal{Q}$ by angular resolution $\psi_\theta = 0.5^\circ$ and temporal resolution $\psi_t = \frac{1}{75}$ seconds of the scanner (as given by Table 2.1). This results in a unit spaced grid, hence the index into each dimension is also their position in this scaled sensor configuration space which we use for convenience in all of our computations. In the absence of a stronger prior this scaling also ensures that each dimension of $\mathcal{Q}$ is treated equally in the regression. For the algorithm in Chapter 6 we are interested in using a simple covariance function of which we choose the Matérn family (as advocated in [103] and discussed in Chapter 4.4) and parametrise it through two hyperparameters: a single length scale l and a process variance σ_p^2 :

$$k(\mathbf{q}, \mathbf{q}')_{\text{matérn}} = \sigma_p^2 \frac{2^{1-v}}{\Gamma(v)} \left(d\sqrt{2v} \right)^v K_v \left(d\sqrt{2v} \right) \quad (5.1)$$

where

$$d(\mathbf{q}, \mathbf{q}') = \sqrt{(\mathbf{q} - \mathbf{q}')^T \begin{bmatrix} (l.\psi_\theta)^2 & \mathbf{0} \\ \mathbf{0} & (l.\psi_t)^2 \end{bmatrix}^{-1} (\mathbf{q} - \mathbf{q}')} \quad (5.2)$$

$\Gamma(z) = \int_0^\infty t^{z-1} \exp(-t) dt$ is the Gamma function, K is the modified Bessel function

of the second kind (see [2]). $\mathbf{q} = [\theta, t]^T$ is a point in sensor configuration space $\mathcal{Q}$ and Equation 5.1 is a restating of Equation 4.54 using the Mahalanobis distance d of Equation 5.2 rather than the Euclidean distance $d = \|\mathbf{q} - \mathbf{q}'\|_{\mathcal{Q}} = \sqrt{(\mathbf{q} - \mathbf{q}')^T (\mathbf{q} - \mathbf{q}')}$. The use of a length scale l affords us explicit control over the otherwise arbitrary smoothness of our regression surface by varying the width of our regression kernel (see Chapter 4.4) which we do so to account for the physical properties of the laser data.

The Matérn class of covariance function encompasses a number of well known covariance functions such as the exponential k_E when $\nu = \frac{1}{2}$ and the squared exponential k_{SE} (or Radial Basis covariance function) as $\nu \rightarrow \infty$ as discussed in Section 4.4.1.2. As suggested in [88], we explored several common choices of $\nu = \{\frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \infty\}$ over a number of workspace sections from the New College dataset [97] (see Chapter 2) that were varied in complexity.¹ Hyperparameters were set through the maximisation of the log Marginal Likelihood (Equation 4.43) with respect to the hyperparameters (using conjugate gradients) for each covariance function assuming a zero mean GP (as is standard in the GP literature [88]). We found that $\nu = \frac{3}{2}$ consistently produced accurate surface reconstructions (according to the reconstruction error as defined in Section 3.5):

$$k(\mathbf{q}, \mathbf{q}')_{\text{matérn } \nu=\frac{3}{2}} = \sigma_p^2 \left(1 + d\sqrt{3}\right) \exp\left(-d\sqrt{3}\right)$$

Although the smoother covariance functions $\nu = \{\frac{5}{2}, \infty\}$ performed well for simple workspaces, they seemed over constrained in complex scenarios, and vice-versa for the very rough exponential covariance function when $\nu = \frac{1}{2}$. In Figure 5.1a we see how the

¹As noted in Section 4.4.1.2 values above $\nu = \frac{7}{2}$ are generally considered indistinguishable with $\nu \rightarrow \infty$.

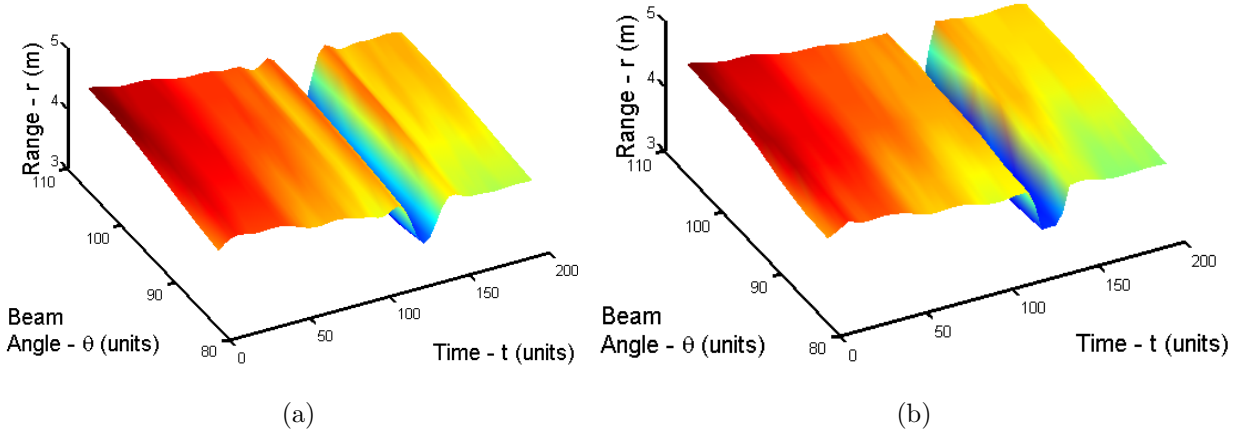


Figure 5.1: Visualisation of GP predictive range surfaces in sensor configuration space for the tree trunk example of Figure 3.7. Each surface is constructed through a triangulation of predictions of laser range measurements $\mathbf{r}^*$ at locations Q^* using two alternative covariance functions: (a) Squared Exponential covariance function k_{SE} and (b) Matérn covariance where $\nu = \frac{3}{2}$, $k_{matérn \nu=\frac{3}{2}}$. Hyperparameters were set by maximising the log Marginal Likelihood of each of the predictive surfaces. $k_{matérn \nu=\frac{3}{2}}$ accurately captures the surface’s abrupt gradient changes that are produced from the tree trunk occluding the college walls, whereas a distinct ringing can be observed for k_{SE} .

strong smoothness property of the squared exponential function k_{SE} leads to ringing around the edge of the tree trunk from Figure 3.9, whereas the $\nu = \frac{3}{2}$ Matérn accurately captures this small, abrupt surface change that is typical in our laser data.

5.3 Is this the Right Covariance Function?

Choosing a covariance function (and their associated hyperparameter values) that will produce consistently high accuracy surface predictions across an entire workspace is a difficult task. This problem is exacerbated by the high variability of underlying workspace surfaces present in our laser data and the infinite choice of covariance functions

that can be constructed from a superposition of any number of simpler functions. That said, we have found that rough covariance functions such as $k_{\text{matérn } \nu=\frac{3}{2}}$ (advocated by Stein [103]) are pertinent to our regression (we provide further analysis into suitable hyperparameter values for our covariance function in Section 6.8.4). By using a single length scale (rather than one for each input dimension) we have assumed a fixed relative scaling between dimensions of $\mathcal{Q}$, in this case of $\frac{\psi_\theta}{\psi_t} = 37.5^\circ \text{s}^{-1}$. On comparing results generated from this simple covariance function (Chapter 6), and those from a multi-length scale covariance function (Chapter 7) however, we find that the accuracy of the surfaces are surprisingly similar (see Chapter 7 for detailed analysis). We also discuss how more complex models may permit increases in accuracy when expert knowledge is available in Section 7.3.4.

Nothing in the following chapters is predicated upon using the Matérn covariance function. In Chapter 6 we provide a Gaussian Process Regression framework that permits the substitution of expedient covariance functions, for example, under different sensor configuration spaces and assumed knowledge. We then extend this framework in Chapter 7 to permit hyperparameter values to vary across the workspace, importantly, in response to local workspace conditions. This provides an element of automatic control over our covariance function, salient to the task at hand. In future works (Section 8.1.2) we discuss how this approach may be applied so that entire covariance function families may also be self selected, thereby eschewing the need to define one a priori.

Chapter 6

Surface Mapping And Decimation In An Active Window Framework

The active window framework mentioned in this chapter was published as ‘Efficient Non-Parametric Surface Representations Using Active Sampling for Push Broom Laser Data’ in the Proceedings of Robotics: Science and Systems, 2010. There was also an invited précis published in the International Joint Conference on Artificial Intelligence, 2011, best paper track.

This chapter concerns the creation of efficient and online surface representations from laser point clouds. We describe an algorithm which simultaneously generates a continuous representation of the workspace from discrete laser samples and decimates the dataset, retaining only locally salient samples. Our method attains decimation factors in excess of two orders of magnitude without significant degradation in fidelity.

The task at hand is the processing of a stream of noisy laser measurements, taken from a planar scanner, to produce a continuous surface representation of a 3D workspace. We emphasise that our surface representation is implicit rather than explicit. We shall arrive at a method which allows us to generate point samples of underlying workspace surfaces at arbitrary locations and densities. The management and propagation of uncertainty is at the heart of this work. It enters our system as measurement noise and is reflected ultimately in the implicit surfaces constructed.

We now present a brief summary of our framework which we shall elucidate further in the Sections 6.3–6.6. We then analyse our approach in Sections 6.7–6.9.

6.1 New Nomenclature Introduced in this Chapter

Active Sampling

$\tilde{\cdot}$	- elements within the current active set
n	- number of measurements in the active set of measurements
η	- span of the active window (index of the newest measurement to have been considered, minus the index of the oldest measurement in the active set of measurements $\tilde{\mathcal{D}}$)
α	- minimum number of measurements between the oldest measurement in $\tilde{\mathcal{D}}$ and online predictions to ensure adequate conditioning
$\mathcal{D}'$	- set of all laser measurements that have been actively selected
$D_{KL}(\cdot \cdot')$	- Kullback-Leibler (KL) divergence
κ	- User selected KL threshold. Sets the trade off between the accuracy of our representations versus the number of retained laser measurements

Parametric Mean GP

$Z(.)$	- zero mean GP regressor $Z(.) \sim \mathcal{GP}(0, k(.,.))$
$\mathbf{h}(.)$	- basis functions $[1, \theta, \theta^2, \theta^3, t, t^2, \theta t, \theta t^2, \theta^2 t, \theta^2 t^2]^T$
e	- number of basis functions
$\boldsymbol{\beta}$	- basis function weightings $\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\beta}; \bar{\mathbf{b}}, B)$
$S(.)$	- parametric mean GP regressor $S(.) = Z(.) + \mathbf{h}(.)^T \boldsymbol{\beta}$
H	- stacked basis functions $[\mathbf{h}(\mathbf{q}_1), \mathbf{h}(\mathbf{q}_2) \dots \mathbf{h}(\mathbf{q}_n)]$
$\bar{\boldsymbol{\beta}}$	- mean of posterior distribution of basis function weightings $\boldsymbol{\beta}$ with respect to ranges. For a uniform $\boldsymbol{\beta}$ ($B^{-1} \rightarrow \mathbf{0}$) $\bar{\boldsymbol{\beta}} = A^{-1} H K_{\mathbf{r}}^{-1} \mathbf{r}$
A	- covariance of posterior distribution of basis function weightings with respect to ranges $A = H K_{\mathbf{r}}^{-1} H^T$ (positive semi-definite)
$\mathbf{u}$	- premultiplier for $\boldsymbol{\beta}$ in parametric GP predictions $\mathbf{u} = \mathbf{h}(\mathbf{q}^*) - H K_{\mathbf{r}}^{-1} \mathbf{k}(Q, \mathbf{q}^*)$

6.2 Algorithm Overview

The use of Gaussian Processes (GPs) in our algorithm (as discussed in Chapter 5) afford non-parametric, probabilistic regression that we use to provide range predictions of our laser scanner via Equation 6.14. However, a well known draw back of GPs is that they scale as $\mathcal{O}(N^3)$ in the number of observations used to condition the predictions. This motivates Section 6.3 where we adopt an *active window* approach whereby our regression uses only a subset of measurements that are in the vicinity of our current

predictions. This allows our algorithm to have constant complexity with respect to the total number of observations in the dataset.

A boon of our approach is that we achieve a compact representation of the underlying workspace surfaces with only a fraction of the laser measurements. For this we use an information theoretic criterion in Section 6.4 to determine which measurements can be predicted from neighbouring measurements in our active window. These measurements are therefore redundant and are not used in our approach. We base our criterion on the Kullback-Leibler (KL) divergence, drawing inspiration from work in sparse GPs [21, 95, 49, 25] (an overview is provided in [87]). Our work is particularly similar to [95] except they use an approximation of the KL divergence and we exploit the ordering of our laser data. The threshold κ , that we place on the KL divergence is used to set the resulting trade off between the accuracy of the representation versus the number of retained laser measurements. Its value is therefore entirely at the user’s discretion.

A property of the smooth GP that we adopt in our framework (as discussed in Section 5.2) is that in areas bereft of data, predictions tend towards the GP’s mean function. It is important that we do not use the commonly adopted “zero mean function” as we need to make predictions at the boundary of the laser scanner’s field of view, and also in places where laser measurements are very sparse. We have found that polynomial surface fitting with only a few terms works effectively for our problem. By following [14, 74] we wrap this polynomial fitting into the overall GP regression to allow our regression to remain non-parametric. The result is a parametric mean GP that we shall discuss in Section 6.5.

To form our active window we exploit the time sequential nature of push broom

laser systems in a fashion which allows our approach to be used online as well as offline. When a new measurement needs to be added to the active window we update our beliefs using just that measurement rather than recomputing from all measurement in the active window. Although this remains the most expensive operation in our algorithm, in Section 6.6 we detail an efficient implementation that reduces our algorithm’s complexity from $\mathcal{O}(n^3)$ in the number of observations in our active window to $\mathcal{O}(n^2)$. This approach is general in that it can be applied to any *stationary* GP (see Section 4.4.1.2) and in Chapter 7 we see how it can form the building blocks for a *non-stationary* variant – that is a GP with a covariance function that can change across the dataset.

We also introduce the notion of *introspection*. Here we allow post-hoc conditioning from laser measurements previously thought to be uninformative. This approach is important in the vicinity of abrupt range discontinuities as we shall demonstrate in Section 6.4. We refer to the resulting process as *active sampling* as the GP framework actively guides us in deciding which of many input measurements should be retained and used later for surface reconstructions.

6.3 Active Windowing

We adopt the parameterisation of Section 3.3 and are interested in estimating a range measurement $r^\star = G(\mathbf{q}^\star)$ from a laser sensor at an arbitrary sensor configuration $\mathbf{q}^\star \in \mathcal{Q}$, where $\mathcal{Q}$ is sensor configuration space and $\mathbf{q}^\star = [\theta^\star, t^\star]^T$ is a vector that contains the query time that the laser scan was taken $t^\star$ and query angle of a laser beam $\theta^\star$. Following Section 4.4 we use Gaussian Process Regression to approximate the mapping

$G(\mathbf{q}^*)$ through a predictive distribution of r^* , $p(r^*|\mathbf{q}^*, \mathcal{D}, \boldsymbol{\xi}) = \mathcal{N}(\mu_S, \sigma_S^2)$, given a model specified by hyperparameters $\boldsymbol{\xi}$, and a set of measurements $\mathcal{D} = \{(\mathbf{q}_i, r_i)\}_{i=1}^N$ (as given by Equation 4.46). For our GP model $S(\cdot)$ we use the Matérn class of smooth covariance function that was detailed in Section 5.2 and a parametric mean function that we shall introduce and discuss in Section 6.5.

As mentioned, the quantities of data we consider render the application of a single monolithic GP infeasible. Instead of using the entire dataset $\mathcal{D}$ (of size N) we enforce a predictive support window $\tilde{\mathcal{D}} = \{(\tilde{\mathbf{q}}_i, \tilde{r}_i)\}_{i=1}^n$ of *fixed* size n for each prediction, where $\tilde{\mathcal{Q}} = \{\tilde{\mathbf{q}}_i\}_{i=1}^n$, $\tilde{\mathbf{r}} = [\tilde{r}_1, \tilde{r}_2 \dots \tilde{r}_n]^T$. This active set $\tilde{\mathcal{D}}$ is formed from the closest (in terms of the order in which the measurements are collected) laser measurements that have been actively accepted by our algorithm. This guarantees a constant time algorithm.

By processing laser data in the same time-sequential order that the measurements are taken we can also use this algorithm online. Here, as the laser sensor progresses along its trajectory the support window slides across sensor configuration space $\mathcal{Q}$. Each time that we wish to adopt a new measurement into $\tilde{\mathcal{D}}$ we first choose to remove the most stale measurement $\min_{t, \theta} \tilde{\mathcal{D}}$ (the measurement at the back of our active window), then expand $\tilde{\mathcal{D}}$ to include the new measurement. We visualise this process in Figure 6.1 with snapshots of the active window taken at various points in time for the tree trunk example (first introduced in Chapter 3). We note in particular that despite a fixed support window size n , that the span η of the active window $\tilde{\mathcal{D}}$ (defined as the index of the newest measurement to have been considered, minus the index of the oldest measurement in $\tilde{\mathcal{D}}$) varies dramatically in response to the changing workspace complexity. This is due to our active sampling technique which is used to select our

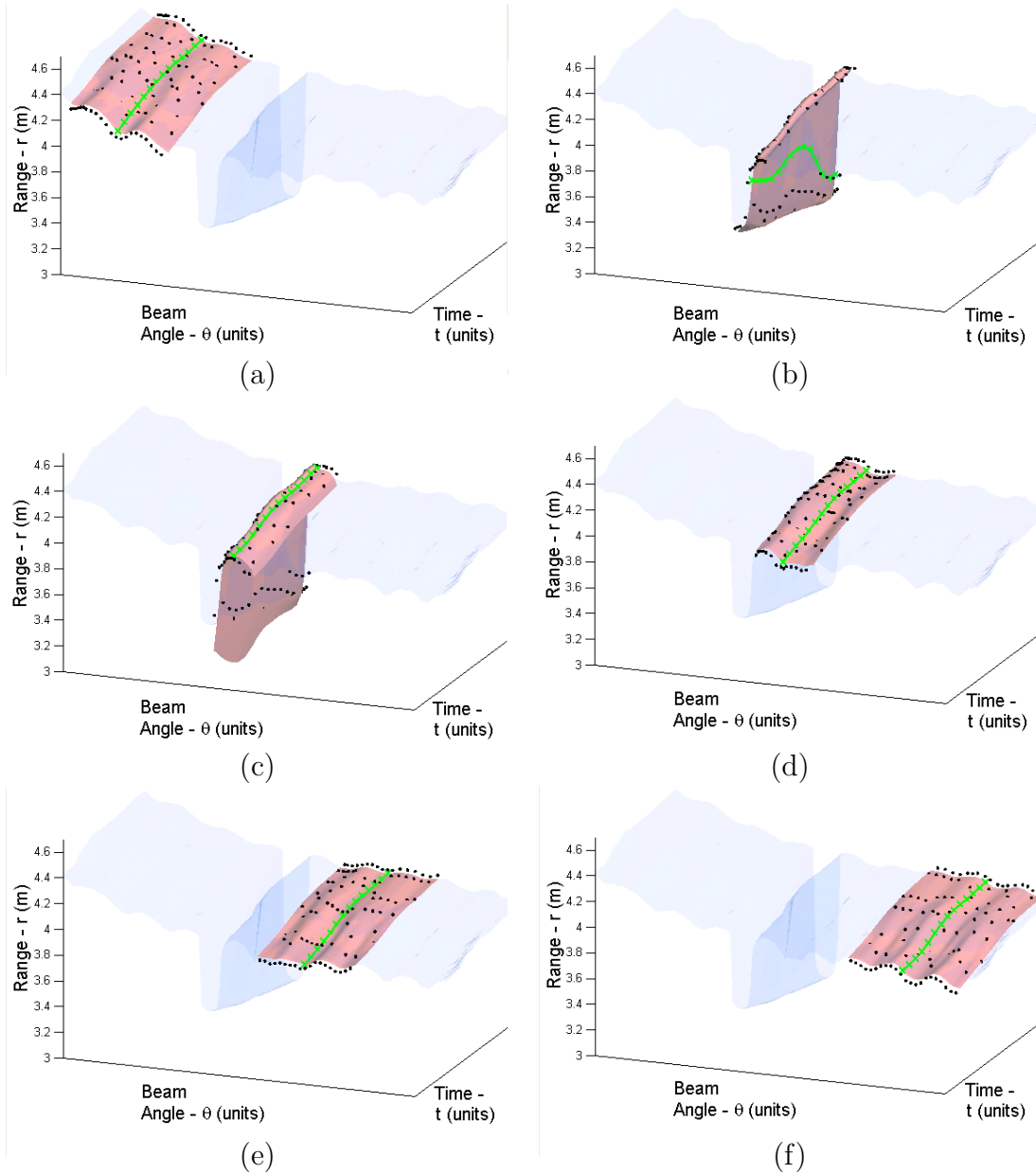


Figure 6.1: Snapshots of our online active window approach for the tree trunk example which was used throughout Chapter 3 (see Figures 3.6 and 3.7). (a) to (f) represent progressive points in time as the active window moves across sensor configuration space $\mathcal{Q}$, denoted by the horizontal axes in the 3D plot. Along the vertical axes are both measured and predicted ranges from the laser sensor to the workspace surfaces. In each plot the blue translucent surface is constructed through linear interpolation all laser measurements $\mathcal{D}$. The red surfaces represent our active window at a given point in time, generated by each prediction of the underlying workspace surface r^* . These predictions are conditioned on the current predictive support window $\tilde{\mathcal{D}}$ (denoted by black dots in the images). The green line represents the mean time $\bar{t}$ of the active window $\tilde{\mathcal{D}}$. Online predictions (Section 6.3.1) are formed whenever $\bar{t}$ passes the prediction query time t^* (Section 6.6.4 contains additional implementation considerations for these predictions). Despite a fixed size of $\tilde{\mathcal{D}}$, n we observe that the span η of the active window can vary dramatically with for example η decreasing surrounding the tree trunk area responsible for the large dip in range measurements. We also note that predictions surrounding the front and back edge vary dramatically as the window progresses, particularly when comparing (c) with (b) and (d).

predictive support window $\tilde{\mathcal{D}}$, the details of which shall be discussed in Section 6.4.

6.3.1 Online Predictions

For our online formulation we use our GP to make predictions $r^* \sim \mathcal{N}(\mu_S, \sigma_S^2)$ both as part of our active sampling technique and for the final predictions of the underlying workspace surfaces that are captured by the laser measurements. Although we are free to make the predictions at any point in sensor configuration space $\mathcal{Q}$, we choose to make these latter predictions only as the mean time $\bar{t}$ (indicated by the green line in Figure 6.1) of the active window $\tilde{\mathcal{D}}$ passes the corresponding t^* in $\mathcal{Q}$. This minimises the risk of forming predictions that fluctuate in response to updates that occur at the front and back of the active window (see the progression of surface visualisations in Figure 6.1b, c and d).

6.3.2 Offline Predictions

The set of all actively selected measurements $\mathcal{D}'$, chosen by an initial pass of our algorithm may also be used for further offline predictions of the underlying workspace surfaces. Each prediction r^* is posed as a query at $\mathbf{q}^*$ for which we re-establish the active support window by using the closest measurements in $\mathcal{D}'$ (in terms of the ℓ_1 norm from sensor configuration $\mathbf{q}$ to our prediction at $\mathbf{q}^*$ in our scaled sensor configuration space $\mathcal{Q}$).¹ Despite a high initial computation cost that is associated with establishing $\tilde{\mathcal{D}}$, further predictions surrounding $\mathbf{q}^*$ are often significantly faster than those made online

¹By maintaining the original ordering of the laser data $\mathcal{D}$ this operation may be approximated by simply using the $\frac{n}{2}$ measurements in either direction from $\mathbf{q}^*$.

as active sampling has already taken place. This is especially true in regions where $\mathcal{D}'$ is very sparse in $\mathcal{Q}$ compared to the original dataset $\mathcal{D}$. Here the same active window $\tilde{\mathcal{D}}$ will span a large proportion of $\mathcal{Q}$ and hence may be re-used for many predictions surrounding $\mathbf{q}^*$. If subsequent queries are in the vicinity of $\mathbf{q}^*$ we may also avoid the initial establishment cost through updating the current $\tilde{\mathcal{D}}$ rather than re-calculating a new $\tilde{\mathcal{D}}$ from scratch as mentioned in the previous section and discussed in detail in Section 6.6.1.

6.4 Active Sampling through Kullback-Leibler Divergence

In this section we describe our method for adaptive laser measurement selection. We decide if a single prediction $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \xi) = \mathcal{N}(\mu_S, \sigma_S^2)$ (Equation 6.14) differs significantly from the corresponding measurement $r_m \sim \mathcal{N}(\mu_m, \sigma_m^2)$ at $\mathbf{q}^*$. We use the Kullback-Leibler (KL) divergence (otherwise known as *information divergence* or *relative entropy*) of the two distributions to ascertain if the average additional information required to specify r_m as a result of using r^* (instead of the true r_m) is greater than a threshold κ . This threshold, naturally modulated by terrain complexity, determines a decimation factor $\frac{|\mathcal{D}|}{|\mathcal{D}'|}$ which we shall discuss further in Section 6.7. Thorough analysis on the choice of all model parameters is presented in Section 6.8 (see Section 7.5 for the non-stationary variant of this algorithm that is discussed in Chapter 7).

The KL divergence measures the distance between two density distributions. For two distributions of a continuous random variable P and V it is given by:

$$D_{KL}(P||V) = \int_{-\infty}^{\infty} p(r) \log_e \left(\frac{p(r)}{v(r)} \right) dr \quad (6.1)$$

where p and v correspond to the densities of P and V . For two 1D Gaussian distributions this has a closed form solution:

$$D_{KL}(\mathcal{N}_m||\mathcal{N}_S) = \frac{1}{2} \left[\log_e \frac{\sigma_S^2}{\sigma_m^2} + \frac{\mu_S^2 + \mu_m^2 - 2\mu_S\mu_m + \sigma_m^2}{\sigma_S^2} - 1 \right] \quad (6.2)$$

where

$$p(r) = \mathcal{N}(\mu_m, \sigma_m^2) \quad (6.3)$$

$$v(r) = \mathcal{N}(\mu_S, \sigma_S^2) \quad (6.4)$$

On adoption of any new laser measurement into the active support set we introspectively analyse the closest (ℓ_1 norm in our scaled sensor configuration space $\mathcal{Q}$) r_m that had previously been deemed redundant to determine if our belief has changed given the new information. On adoption of this measurement, introspection is repeated for the next closest redundant r_m until there are no further adoptions. This approach is demonstrated in Figure 6.2.

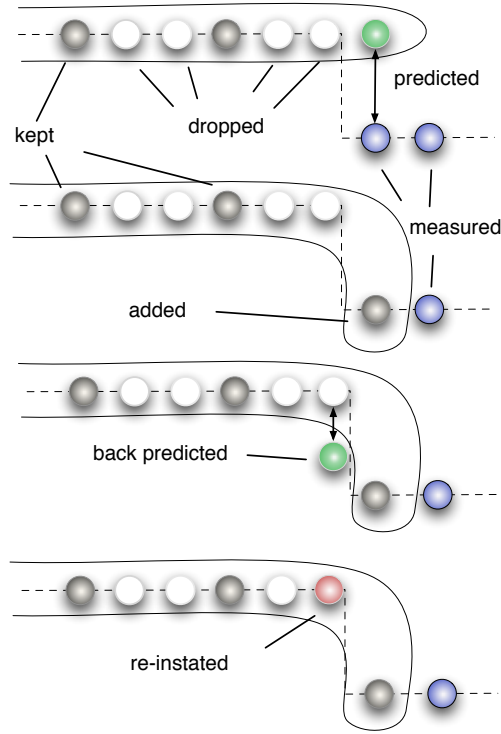


Figure 6.2: Active sampling of a 1D data series. Each node, connected by the chained lines, represents a measurement. Black indicates data accepted by the forward pass of the active sampling algorithm while the white have been rejected and the blue are yet to be observed. Green nodes are predictions made by our algorithm and red are measurements accepted through introspection. Progressing downwards, each subplot represents the algorithm at successive points in time. (top) The algorithm forms a prediction r^* that has a KL divergence from the measurement at the front of the active set r_m that is greater than κ , hence it is adopted into the active support set (2nd row). (3rd row) The algorithm makes a back prediction which, where it was previously in agreement with the measurement, now has a significant error, that is greater than κ . (bottom) The measurement, which was previously rejected, is therefore accepted into the active region.

Introspection allows both sides of discontinuities to be analysed given the final sampling of the opposite side of the discontinuity. Figure 6.3 demonstrates the results of this reverse sweeping, and the typical $\mathcal{D}'$ that is stored and used for subsequent predictions

instead of $\mathcal{D}$. In the worst case, where there are truly complex regions of the workspace, our algorithm performs as well as a naïve implementation by using all available $\mathcal{D}$ in that region, while maintaining the ability to automatically subsample simple scenes. We note that the resulting subsampling may also provide an information rich subset of $\mathcal{D}$. This may boost performance of other applications especially registration, where careful selection of laser points can increase accuracy and robustness (see discussion in [92] and Appendix C).

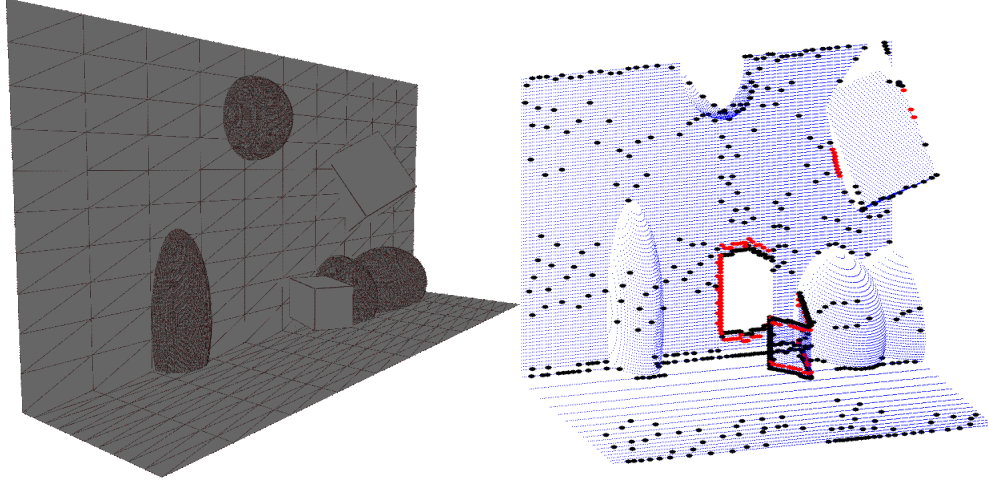


Figure 6.3: Result from using active sampling within our sensor configuration space formulation. (Left) CAD model used to generate synthetic data based on the wall section ($7 \times 3 \times 1.5$ m) of test case (a) (see Figure 3.10) with added objects. (Right) Plot of synthetic data as a laser scanner is dragged from left to right in front of the scene. Blue indicates all laser measurements, black denotes measurements selected on the forward pass of the algorithm, and red are measurements that have been chosen introspectively. We have used $\kappa = 0.8$ nats as in other examples (see Figures 6.4 and 6.9). Note that the floating cube in the centre of the scene generates a large (1 m) discontinuity which is automatically highly sampled on both sides. For a plot indicating the effect of varying κ see Figure 6.6.

6.5 Parametric Mean Function

We affect a parametric non-zero GP mean function $\mu(\mathbf{q})$ through the use of explicit basis functions, $\mathbf{h}(\mathbf{q})$ with weightings $\boldsymbol{\beta} \sim \mathcal{N}(\bar{\mathbf{b}}, B)$ which are commonly found in a traditional Bayesian weight space view of linear parametric regression (see [88] and Section 4.3). These basis functions are combined with a zero-mean GP $Z(\mathbf{q}) \sim \mathcal{GP}(0, k(\mathbf{q}, \mathbf{q}') + \delta(\mathbf{q}, \mathbf{q}'))$ of Section 4.4

$$S(\mathbf{q}) = Z(\mathbf{q}) + \mathbf{h}(\mathbf{q})^T \boldsymbol{\beta} \quad (6.5)$$

$$S(\mathbf{q}) \sim \mathcal{GP}(\mathbf{h}(\mathbf{q})^T \bar{\mathbf{b}}, k(\mathbf{q}, \mathbf{q}') + \mathbf{h}(\mathbf{q})^T B \mathbf{h}(\mathbf{q}')) \quad (6.6)$$

In the absence of data, predictions using $S(\mathbf{q}^*) = p(r^* | \mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi})$ tend to the surface estimates of the explicit basis functions, rather than that of the zero mean GP $Z(\mathbf{q}^*)$ (assuming a smooth covariance function such as the one defined in Section 5.2). One may view this formulation as the GP modelling the residuals between the implicit surface generated by $\mathbf{h}(\mathbf{q})^T \boldsymbol{\beta}$, and $\tilde{\mathcal{D}}$. This approach has been used successfully in robotics applications such as [70] and is particularly beneficial when predicting near the boundary of our active window. It is the key to achieving our compression rates.²

To extract the posterior predictive distribution under this model we follow the same derivation as Section 4.4, but we must perform an additional marginalisation for the nuisance parameter weightings $\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\beta}; \bar{\mathbf{b}}, B)$. The important equations here are the

²One may view this formulation from a frequency domain viewpoint whereby the mean function captures the low frequency components that are present throughout most of the workspace. The majority of the laser data are required to capture the high frequency signals that surround only complex regions of the workspace.

definition of the GP used in this chapter in Equation 6.5 and the predictions that can be made from it, which we shall arrive at in Equation 6.14. The following derivation is provided for completeness but we suggest skipping to the end of this section on the first read through of this chapter.

In Equation 4.45 we formed the joint distribution between training data (in this case $\tilde{\mathcal{D}} = \{(\tilde{\mathbf{q}}_i, \tilde{r}_i)\}_{i=1}^n$) and prediction r^* at location $\mathbf{q}^*$ for a GP with explicit mean function $\boldsymbol{\mu}(Q)$. For the GP defined in Equation 6.5 this joint distribution may be written as

$$p(\tilde{\mathbf{r}}, r^* | \mathbf{q}^*, \tilde{Q}, \boldsymbol{\xi}, \boldsymbol{\beta}) = \mathcal{N} \left(\begin{bmatrix} \tilde{\mathbf{r}} \\ r^* \end{bmatrix} ; \begin{bmatrix} H^T \boldsymbol{\beta} \\ \mathbf{h}^T(\mathbf{q}^*) \boldsymbol{\beta} \end{bmatrix}, \begin{bmatrix} K_{\tilde{\mathbf{r}}} & \mathbf{k}(\tilde{Q}, \mathbf{q}^*)^T \\ \mathbf{k}(\tilde{Q}, \mathbf{q}^*) & k(\mathbf{q}^*, \mathbf{q}^*) + \sigma_m^2 \end{bmatrix} \right) \quad (6.7)$$

where

$$H = [\mathbf{h}(\mathbf{q}_1), \mathbf{h}(\mathbf{q}_2) \dots \mathbf{h}(\mathbf{q}_n)] \quad (6.8)$$

$$\mathbf{h} = [1, \theta, \theta^2, \theta^3, t, t^2, \theta t, \theta t^2, \theta^2 t, \theta^2 t^2]^T \quad (6.9)$$

$K_{\tilde{\mathbf{r}}}$ is $K_{\mathbf{r}}$ from Equation 4.44 but constructed with only measurements in the active set $\tilde{\mathcal{D}}$. On conditioning our predictions on our training data (through the application of the posterior distribution in Equation D.16) we arrive at

$$p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi}, \boldsymbol{\beta}) = \mathcal{N}\left(r^*; \mathbb{E}[r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi}, \boldsymbol{\beta}], \mathbb{V}[r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi}, \boldsymbol{\beta}]\right) \quad (6.10)$$

$$\begin{aligned} \mathbb{E}[r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi}, \boldsymbol{\beta}] &= \mathbf{h}^T(\mathbf{q}^*)\boldsymbol{\beta} + \mathbf{k}(\tilde{Q}, \mathbf{q}^*)^T K_{\tilde{\mathbf{r}}}^{-1} (\tilde{\mathbf{r}} - H^T \boldsymbol{\beta}) \\ &= \mathbb{E}\left[r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi}\right]_{\mu(\mathbf{q})=0} + \mathbf{u}^T \boldsymbol{\beta} \end{aligned} \quad (6.11)$$

$$\begin{aligned} \mathbb{V}[r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\beta}, \boldsymbol{\xi}] &= k(\mathbf{q}^*, \mathbf{q}^*) + \sigma_m^2 - \mathbf{k}(\tilde{Q}, \mathbf{q}^*)^T K_{\tilde{\mathbf{r}}}^{-1} \mathbf{k}(\tilde{Q}, \mathbf{q}^*) \\ &= \mathbb{V}\left[r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi}\right]_{\mu(\mathbf{q})=0} \end{aligned} \quad (6.12)$$

where we have gathered all terms concerning $\boldsymbol{\beta}$ into $\mathbf{u}$

$$\mathbf{u} = \mathbf{h}(\mathbf{q}^*) - H K_{\tilde{\mathbf{r}}}^{-1} \mathbf{k}(\tilde{Q}, \mathbf{q}^*) \quad (6.13)$$

The terms $\mathbb{E}\left[r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi}\right]_{\mu(\mathbf{q})=0}$ and $\mathbb{V}\left[r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi}\right]_{\mu(\mathbf{q})=0}$ are the expectation and variance of the zero mean $Z(\mathbf{q}^*)$ GP (see Equations 4.47 and 4.48 respectively). Noting that Equation 6.10 is a linear function of $\boldsymbol{\beta}$ and the posterior distribution in $\boldsymbol{\beta}$, $p(\boldsymbol{\beta}|\tilde{\mathcal{D}}) = \mathcal{N}(\boldsymbol{\beta}; \bar{\boldsymbol{\beta}}, A^{-1})$ is given by Equation 4.26 we may then marginalise $\boldsymbol{\beta}$ through Equation 4.14

$$S(\mathbf{q}^*) = p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi}) = \mathcal{N}\left(r^*; \mathbb{E}\left[r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi}\right]_{\mu(\mathbf{q})=0} + \mathbf{u}^T \bar{\boldsymbol{\beta}}, \mathbb{V}\left[r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi}\right]_{\mu(\mathbf{q})=0} + \mathbf{u}^T A^{-1} \mathbf{u}\right) \quad (6.14)$$

where

$$A = \lim_{B^{-1} \rightarrow \mathbf{0}} (B^{-1} + HK_{\tilde{\mathbf{r}}}^{-1}H^T) = HK_{\tilde{\mathbf{r}}}^{-1}H^T \quad (6.15)$$

$$\bar{\boldsymbol{\beta}} = \lim_{B^{-1} \rightarrow \mathbf{0}} (A^{-1} (H\Omega^{-1}\tilde{\mathbf{r}} + B^{-1}\bar{\mathbf{b}})) = A^{-1}HK_{\tilde{\mathbf{r}}}^{-1}\tilde{\mathbf{r}} \quad (6.16)$$

$\bar{\boldsymbol{\beta}}$ was reported in Equation 4.28 and we may freely substitute covariance $K_{\tilde{\mathbf{r}}}$ for the covariance between measurements in our parametric regression model Ω of Equation 4.25 due to the consistency property of Gaussian Distributions (see Appendix D). We take the limiting case as the prior on the weightings tends towards a uniform distribution, ($B^{-1} \rightarrow \mathbf{0}$). As noted by, amongst others [14, 74], we arrive at a predictive distribution that is independent of $\mathbf{b}$. This is a reassuring result as $\bar{\boldsymbol{\beta}}$ is simply the weighted linear least squares fit of $\boldsymbol{\beta}$. This parametric mean GP formulation affords us the benefit of explicit basis functions without requiring additional parameters, and maintains the desirable predictive properties of GPs.

In addition to the standard GP equation (Equation 4.46) an inverse, A^{-1} , must be calculated but its size is that of the number of explicit basis functions chosen for the application e , and not the support set size, n . We choose the 10 polynomial basis functions as given in Equation 6.9 and have found that cross terms above second order have marginal overall impact on our predictions.

6.6 Implementation Considerations

This section discusses some of the important considerations that are required to create a practical implementation of our algorithm. As advocated in Chapter 4.4.2 we use

Cholesky decomposition to mitigate the undesirable effects of ill conditioning (indicated by a high condition number - the ratio of the largest eigenvalue to the smallest) during the inversion of matrices. Specifically we create upper triangular Cholesky Factors for both symmetric positive semidefinite matrices, defined in Equations 4.44 and 6.15

$$K_{\tilde{\mathbf{r}}} = C_{\tilde{\mathbf{r}}}^T C_{\tilde{\mathbf{r}}} \quad (6.17)$$

$$A = F^T F \quad (6.18)$$

These Cholesky Factors $C_{\tilde{\mathbf{r}}}$ and F are computed in $\mathcal{O}(n^3)$ and $\mathcal{O}(e^3)$ (where n is the number of elements in the active set $\tilde{\mathcal{D}}$ and e the number of explicit basis functions $\mathbf{h}$), and are used to efficiently solve triangular sets of linear equations as summarised in Table 6.2 (see Section 4.4.2 for details). By using these precomputed elements equations such as Equation 6.15 simplify to

$$A = H K_{\tilde{\mathbf{r}}}^{-1} H^T = H \left(C_{\tilde{\mathbf{r}}} \setminus (C_{\tilde{\mathbf{r}}}^T \setminus H^T) \right) = \Pi^T \Pi \quad (6.19)$$

We also seek to maximise the speed and accuracy of our surface predictions. To this end we investigate techniques to expedite our implementation which exploit the special structure of our Cholesky Factors $C_{\tilde{\mathbf{r}}}$ and F in Sections 6.6.1 and 6.6.2. For a given computational capacity these techniques permit us the use a larger active set size n than otherwise possible, allowing predictions to be more fully conditioned, both while $\mathcal{D}'$ is constructed online and for subsequent surface querying. To roughly gauge the computational requirements of this algorithm see the timing diagrams in Section

Quantity	Used Instead	In Equations
$K_{\tilde{\mathbf{r}}}^{-1}\tilde{\mathbf{r}}$	$\mathbf{v} = C_{\tilde{\mathbf{r}}}^T \backslash \tilde{\mathbf{r}}$ (6.20)	4.43, 4.47, 6.16, 7.6
$K_{\tilde{\mathbf{r}}}^{-1}\mathbf{k}(\tilde{Q}, \mathbf{q}^*)$	$\boldsymbol{\lambda} = C_{\tilde{\mathbf{r}}}^T \backslash \mathbf{k}(\tilde{Q}, \mathbf{q}^*)$ (6.21)	4.47, 4.48, 6.13
$K_{\tilde{\mathbf{r}}}^{-1}H^T$	$\Pi = C_{\tilde{\mathbf{r}}}^T \backslash H^T$ (6.22)	6.16, 6.13, 6.15
$A^{-1}HK_{\tilde{\mathbf{r}}}^{-1}\tilde{\mathbf{r}}$	$\boldsymbol{\epsilon} = F^T \backslash \Pi^T \mathbf{v}$ (6.23)	6.14, 6.16, 6.5

Table 6.2: Summary of alterations to equations used in our windowed framework on account of implementation considerations. $\backslash$ has been used to indicate where forward substitution is used as detailed in Section 4.4.2. Each substitution contains only one Cholesky factor of Equations 6.17 and 6.18 and hence must be combined as per Equation 6.19.

6.8.3. In Sections 6.6.3 and 6.6.4 we then make two practical considerations to ensure that our predictions are accurate even while predicting the most demanding underlying workspace surfaces.

6.6.1 Updating Cholesky Factors

After the initial Cholesky Factor $C_{\tilde{\mathbf{r}}}$ is established (an $\mathcal{O}(n^3)$ operation) we may drop an order of complexity for subsequent calculations of $C_{\tilde{\mathbf{r}}}$ through intelligent re-use of its elements. This is a common technique [93, 94, 78] and reduces the complexity of our algorithm to $\mathcal{O}(n^2)$. To do so we allow only one measurement to enter or leave the active set $\tilde{\mathcal{D}}$ at a time and exploit the form of the Cholesky decomposition algorithm. Specifically, each element of the upper triangular Cholesky Factor $C_{\tilde{\mathbf{r}}_{i,j}}$ (where i and j are row and column indicies) is constructed from the corresponding element $K_{\tilde{\mathbf{r}}_{i,j}}$ and neighbouring Cholesky elements: above ($C_{\tilde{\mathbf{r}}_{k,j}}$, $k < j$), on the leading diagonal to the left ($C_{\tilde{\mathbf{r}}_{i,i}}$) and above this leading diagonal element ($C_{\tilde{\mathbf{r}}_{k,i}}$, $k < i$). Therefore to

insert or marginalise measurements (by inserting or removing the corresponding rows and columns of $K_{\tilde{\mathbf{r}}}$) one can reuse all elements that are above the corresponding row in the old Cholesky Factor. The update of elements of $C_{\tilde{\mathbf{r}}}$ to the right and below the insertion or deletion are more complicated. This requires the addition or subtraction of the outer product of the corresponding Cholesky factor elements (see $R_{3,3}$ in Equation 6.25 and $C_{3,3}$ in Equation 6.27). Fortunately, for rank one updates (corresponding to the addition or marginalisation of a single measurement) there exist a number of algorithms to efficiently compute this, for a review see [31]. We choose to use maybe the most stable technique by Stewart which is widely available in the LINPACK library [27] and has complexity $\mathcal{O}(n^2)$. This procedure may be applied multiple times to add and remove multiple measurements as desired.

Whenever we wish to accept a new measurement into our active set $\tilde{\mathcal{D}}$ we first remove the most stale measurement $\min_{t,\theta} \tilde{D}$ (the measurement at the back of our active window). We achieve this through concatenation of the marginalisation and expansion operations (setting elements in the following block matrices to zero as appropriate). Both operations are performed as an atomic unit to avoid unnecessary matrix copying through storing intermediate results in place in $C_{\tilde{\mathbf{r}}}$.

Marginalisation

Given a multivariate normal distribution $\tilde{\mathbf{r}} \sim \mathcal{N}(\tilde{\mathbf{r}}; \boldsymbol{\mu}, K_{\tilde{\mathbf{r}}})$ with positive semi-definite covariance matrix $K_{\tilde{\mathbf{r}}}$, and its upper triangular Cholesky factor $C_{\tilde{\mathbf{r}}}$ (Equation 6.17)

$$\tilde{\mathbf{r}} = \begin{bmatrix} \tilde{\mathbf{r}}_1 \\ \tilde{r}_2 \\ \tilde{\mathbf{r}}_3 \end{bmatrix} \quad K_{\tilde{\mathbf{r}}} = \begin{bmatrix} K_{1,1} & \mathbf{k}_{1,2} & K_{1,3} \\ \mathbf{k}_{1,2}^T & k_{2,2} & \mathbf{k}_{2,3}^T \\ K_{1,3}^T & \mathbf{k}_{2,3} & K_{3,3} \end{bmatrix} \quad C_{\tilde{\mathbf{r}}} = \begin{bmatrix} C_{1,1} & \mathbf{c}_{1,2} & C_{1,3} \\ 0 & c_{2,2} & \mathbf{c}_{2,3}^T \\ 0 & 0 & C_{3,3} \end{bmatrix} \quad (6.24)$$

we marginalise (remove) measurement $\tilde{r}_2$ through removal of the central row and column of the block matrix $K_{\tilde{\mathbf{r}}}$ to produce $K'_{\tilde{\mathbf{r}}}$. The resulting Cholesky Factor $R_{\tilde{\mathbf{r}}}$ is related to $C_{\tilde{\mathbf{r}}}$ via:

$$R_{\tilde{\mathbf{r}}} = \begin{bmatrix} R_{1,1} & R_{1,3} \\ 0 & R_{3,3} \end{bmatrix} = \begin{bmatrix} C_{1,1} & C_{1,3} \\ 0 & \Upsilon(C_{3,3}^T C_{3,3} + \mathbf{c}_{2,3} \mathbf{c}_{2,3}^T) \end{bmatrix} \quad (6.25)$$

Where Υ is the Cholesky update of [27]. An efficient implementation can be found in, for example, *cholupdate* of MATLAB® [61].

Expansion

Given positive semi-definite covariance matrix $K'_{\tilde{\mathbf{r}}}$ (for the multivariate Gaussian distribution of $\tilde{\mathbf{r}}'$) and its corresponding upper triangular Cholesky factor $R_{\tilde{\mathbf{r}}}$ (Equation 6.25),

$$\tilde{\mathbf{r}}' = \begin{bmatrix} \tilde{\mathbf{r}}_1 \\ \tilde{\mathbf{r}}_3 \end{bmatrix} \quad K'_{\tilde{\mathbf{r}}} = \begin{bmatrix} K_{1,1} & K_{1,3} \\ K_{1,3}^T & K_{3,3} \end{bmatrix} \quad (6.26)$$

using the definitions of Equation 6.24 we may insert a central row and column into $K'_{\tilde{\mathbf{r}}}$ to retrieve $K_{\tilde{\mathbf{r}}}$ which correspond to additional measurement $\tilde{r}_2$ by expanding $R_{\tilde{\mathbf{r}}}$ to $C_{\tilde{\mathbf{r}}}$ where

$$C_{\tilde{\mathbf{r}}} = \begin{bmatrix} R_{1,1} & R_{1,1}^T \setminus \mathbf{k}_{1,2} & R_{1,3} \\ 0 & \sqrt{k_{2,2} - \mathbf{c}_{1,2}^T \mathbf{c}_{1,2}} & \frac{\mathbf{k}_{2,3}^T - \mathbf{c}_{1,2}^T C_{1,3}}{c_{2,2}} \\ 0 & 0 & \Upsilon(R_{3,3}^T R_{3,3} - \mathbf{c}_{2,3} \mathbf{c}_{2,3}^T) \end{bmatrix} \quad (6.27)$$

Computation must be ordered from top left to bottom right so that intermediate results for each block matrices can be used in subsequent calculations. We have used $\setminus$ to indicate where forward substitution is used to solve for additional Cholesky Factor elements which are as a result of the insertion.

6.6.2 Updating Forward Substitutions

We can also update the forward solved variable $\mathbf{v}$ (Equation 6.20) in light of the replacement of $\tilde{r}_2$, again saving us from having to recompute from scratch each time. We must note however that the update of $\boldsymbol{\lambda}$ (Equation 6.21) is not possible due to the dependence on $\mathbf{k}(\tilde{Q}, \mathbf{q}^*)$ which changes for each prediction. Although there is nothing stopping us in theory from also efficiently updating Π in Equations 6.22 through treating each column of H^T separately, we find that in practise this is not feasible as we shall discuss in Section 6.6.3.

Marginalisation

We wish to update the solution $\mathbf{v} = C_{\tilde{\mathbf{r}}}^T \setminus \tilde{\mathbf{r}}$ (Equation 6.20) under the marginalisation (removal) of a measurement $\tilde{r}_2$ from the multivariate normal distribution $\tilde{\mathbf{r}} \sim \mathcal{N}(\tilde{\mathbf{r}}; \boldsymbol{\mu}, K_{\tilde{\mathbf{r}}})$, of which $C_{\tilde{\mathbf{r}}}$ is the upper triangular Cholesky Factor of semi-positive definite covariance function $K_{\tilde{\mathbf{r}}}$.

$$\begin{bmatrix} C_{1,1}^T & 0 & 0 \\ \mathbf{c}_{1,2}^T & c_{2,2} & 0 \\ C_{1,3} & \mathbf{c}_{2,3} & C_{3,3}^T \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ v_2 \\ \mathbf{v}_3 \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{r}}_1 \\ \tilde{r}_2 \\ \tilde{\mathbf{r}}_3 \end{bmatrix} \quad (6.28)$$

We therefore wish to find $\mathbf{w}$ where $R_{\tilde{\mathbf{r}}}$ corresponds to the contracted Cholesky Factor of $C_{\tilde{\mathbf{r}}}$ as in Equation 6.25:

$$\begin{bmatrix} R_{1,1}^T & 0 \\ R_{1,3}^T & R_{3,3}^T \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_3 \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{r}}_1 \\ \tilde{\mathbf{r}}_3 \end{bmatrix} \quad (6.29)$$

where

$$\mathbf{w} = \begin{bmatrix} \mathbf{v}_1 \\ R_{3,3}^T \setminus (\tilde{\mathbf{r}}_3 - R_{1,3}^T \mathbf{v}_1) \end{bmatrix} \quad (6.30)$$

Expansion

To insert measurement $\tilde{r}_2$ into the system of linear equations $\mathbf{w} = R_{\tilde{\mathbf{r}}}^T \setminus \tilde{\mathbf{r}}'$ (6.29) to retrieve $\mathbf{v} = C_{\tilde{\mathbf{r}}}^T \setminus \tilde{\mathbf{r}}$ (Equation 6.28) given $R_{\tilde{\mathbf{r}}}$, $C_{\tilde{\mathbf{r}}}$ and $\mathbf{w}$ we require that $\mathbf{v}$ is given by

$$\mathbf{v} = \begin{bmatrix} \mathbf{w}_1 \\ \frac{\tilde{r}_2 - \mathbf{c}_{1,2}^T \mathbf{w}_1}{c_{22}} \\ C_{3,3}^T \setminus (R_{3,3}^T \mathbf{w}_3 - \mathbf{c}_{2,3} v_2) \end{bmatrix} \quad (6.31)$$

where calculations are performed in order from top to bottom so that intermediate results for v_2 can be used to construct $\mathbf{v}_3$.

6.6.3 Mean Function Conditioning

Although we use the Cholesky decomposition of A (Equation 6.15) and use only low order polynomial basis functions $\mathbf{h}$ (Equation 6.8) in its construction, the conditioning of A can still become prohibitive when this algorithm is applied to very large datasets and when very smooth covariance functions are used (see Chapter 4.4.2). Fortunately, the only non-stationary element (variant to translations in the input space) of our parametric GP (Section 6.5) is the set of polynomial basis functions $\mathbf{h}$. The further that these basis functions $\mathbf{h}$ are evaluated from the origin the more highly correlated they become and hence the worse the resulting condition number of A . We can therefore reduce the condition number of A by applying a shift in $\mathcal{Q}$ for the construction of H whilst leaving the rest of our algorithm unchanged. Specifically we use H' to replace H in all equations:

$$H'(\tilde{Q}) = H(\tilde{Q} - \bar{\mathbf{q}}) \quad (6.32)$$

where

$$\bar{\mathbf{q}} = [\bar{\theta}, \bar{t}]^T \quad (6.33)$$

$\bar{\theta}$ and $\bar{t}$ are the means of the current active set in each dimension of $\mathcal{Q}$.

An efficient extension could be to update $\bar{\mathbf{q}}$ only when the conditioning of A becomes poor. Between each update of $\bar{\mathbf{q}}$ elements of H' that remain within the active set $\tilde{\mathcal{D}}$ would no longer require re-calculation on adoption of new measurements. It may then also be efficient to update rather than recalculate the dependent variable Π (Equation 6.22) following the strategy detailed in Section 6.6.2. We return to this discussion and

a practical implementation in Section 8.1.1 as future work.

6.6.4 Forming Predictions during Introspection

We aim to form our online predictions of the underlying workspace surfaces after each round of introspection. This is so that we fully capture both sides of any discontinuities as explained in Section 6.4. On occasions when the span of the active window η collapses dramatically however (for example on transition from planar surfaces to a highly complex workspace such as foliage), we are forced to make predictions before an introspective pass has finished. This is so that we avoid making predictions near the front or back of the active set $\tilde{\mathcal{D}}$ where there is less support for predictions, as per our prediction strategy (see Section 6.3.1 and Figure 6.1). For every step of introspection we therefore make predictions prematurely if the time of the α th measurement from the back of the active set t_α is younger than the last time that a prediction was made t_Δ^* ($t_\alpha > t_\Delta^*$).³

6.7 Experimental Setup

In our analysis we aim to investigate how changing the KL Divergence threshold κ affects the compression rate of point clouds collected in a commonplace urban environment. We use the real world examples presented in Chapter 3.5 (the images are reproduced along the top column of Figure 6.4) as well as synthetic data, which we have generated from two CAD models (test cases (e) and (f)). Test case (e) is a simplification of test

³Note: this consideration is only required for online predictions; offline predictions are guaranteed to be post active sampling (see Section 6.3.2).

case (a) where we have represented the wall and floor by vertical and horizontal planes of the same proportions as in the real world, and (for simplicity) approximate the laser scanners trajectory as a single straight line that runs parallel to the wall. A more complex model is used for test case (f) (Figure 6.3), based on the one used for test case (e) but with added objects in the scene.

Our success metrics are the reconstruction error (in metres) that was introduced in Chapter 3.12 and compression rate $\frac{|\mathcal{D}'|}{|\mathcal{D}|}$ %. Specifically we define error as the ℓ_1 norm in range between a hold out set of laser measurements and predictions, made online as $\mathcal{D}'$ is established. Compression rate is defined as the percentage of the laser measurements that are used in our representation over the total number of laser measurements. Therefore a high decimation factor $\frac{|\mathcal{D}|}{|\mathcal{D}'|}$ (and a high compression) achieves a low compression rate. We show that as KL divergence threshold κ increases, that the average error increases and the compression rate decreases.

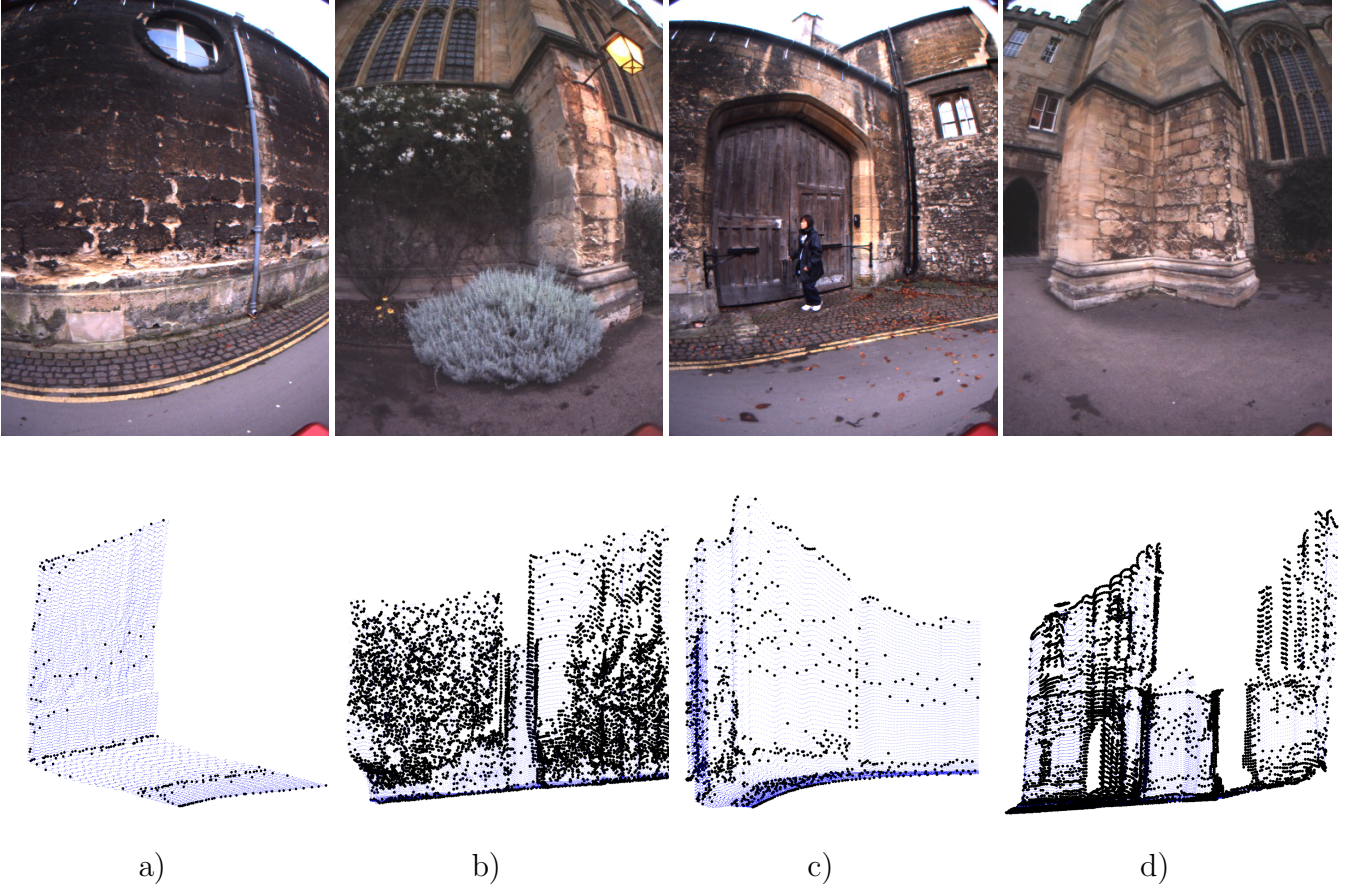


Figure 6.4: Real-world test cases (column-wise). Images (Top) were captured at the same time as the laser data displayed in Euclidean space (bottom). Blue points represent the original laser measurements $\mathcal{D}$. Black indicates the actively sampled measurements $\mathcal{D}'$ for $\kappa = 0.8 \text{ nats}$, as detailed in Section 6.8. (Note that, for ease of interpretation of the scene, images and scans are not exactly aligned and sometimes vary in aspect angle and field of view relative to each other.) Our approach retains the majority of data in salient regions of the workspace $\mathcal{W}$ such as at the intersections of planes, around the outline of a person (to the left in the scan in part (c)), on bushes and other complex geometries.

6.8 Results

We analyse compression rate in Section 6.8.1, resulting reconstruction error in Section 6.8.2 and computation time in Section 6.8.3 – all as we vary our prediction threshold κ . Throughout we have used the online variant of our algorithm (see Section 6.3.1) and we use the Matérn class of covariance function with hyperparameter $\xi = \{l, \sigma_p^2\}$ (see Section 3.4). For this we use length scale $l = 8$ units and process variance $\sigma_p^2 = 0.05 \text{ m}^2$ based on a sensitivity analysis that will be given in Section 6.8.4. We have also chosen to use measurement variance $\sigma_m^2 = 0.01 \text{ m}^2$ (see Chapter 4) and explore the effect of varying this regression model in Section 6.8.5. We have also used an active window size $n = 200$ (see Section 6.3) and $\alpha = 5$ (see Section 6.6.4) which were found empirically to produce consistently accurate surface representations for a given compression rate $\frac{|\mathcal{D}'|}{|\mathcal{D}|} \%$. For $n < 200$ and high divergence thresholds κ the active window would typically span too few laser scan lines to produce predictions that had sufficient support in the time t dimension (see Section 6.6.4).

6.8.1 Compression Rate

Figure 6.5 depicts typical subsampling results as κ is varied. For aggressive compression rate $\frac{|\mathcal{D}'|}{|\mathcal{D}|} \%$ of 0.1 % we achieve a mean error of 0.3 m. For mean errors comparable with measurement precision of 0.015 m we can achieve a decimation factor of 6.

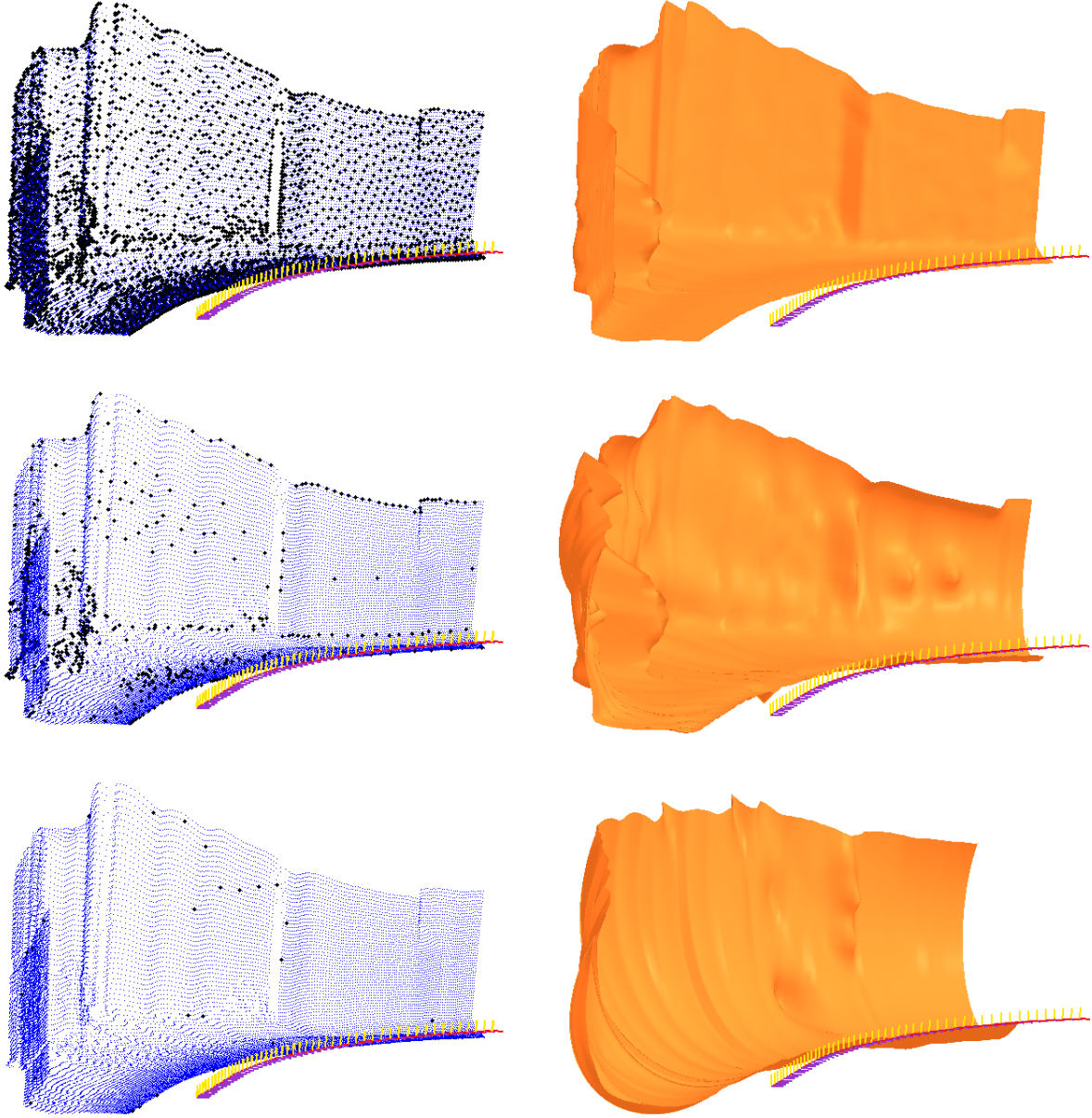


Figure 6.5: Laser measurement subsample plots for various values of our KL divergence threshold κ for test case (c) of Figure 6.4, plotted in Euclidean - x, y, z space. Each row corresponds to $\kappa = \{0.3, 1.26, 3.18\}$ nats working downwards. (Left) the original 28892 measurements are in blue and measurements selected by our active sampling algorithm are in black. (Right) surface representation using only the measurements selected by our algorithm. The surface is constructed from the mean of the predictions from our algorithm in a regular grid online, as discussed in in Section 6.6.4. These predictions are projected into Euclidean space via the trajectory of the laser scanner (represented by the coloured frames across the bottom of each plot). (Top) 4543 laser measurements are used to produce predictions resulting in a mean error of 0.014m while in (Middle) 323 measurements are used with a 0.177 m average error. (Bottom) only 28 measurements have been selected, providing an average error of 0.32 m.

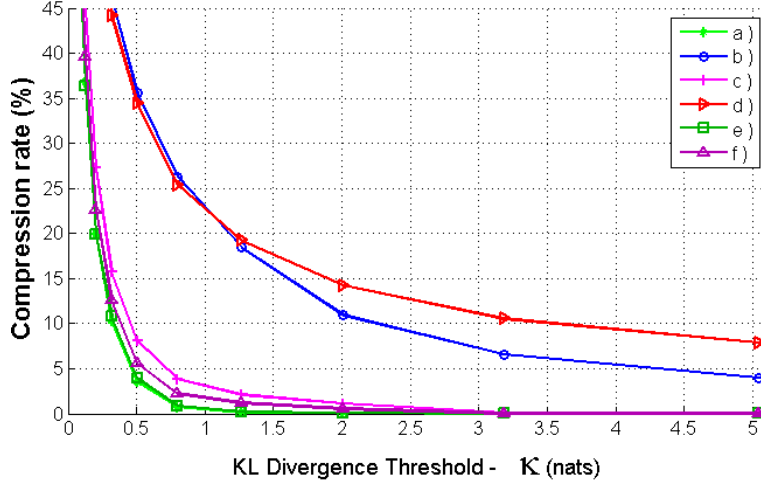


Figure 6.6: Compression rate versus κ for a variety of urban and synthetic environments. Compression rate, measured as the percentage of $\mathcal{D}$ selected by our algorithm ($\frac{|\mathcal{D}'|}{|\mathcal{D}|} \%$), is plotted against κ at each of the configurations denoted by markers. A lower compression rate results from higher thresholds and scenes that are less complex. See Section 6.7 for a discussion of the test cases.

In Figure 6.6 results are collated across a range of κ for the six test cases. Intuitively, as the threshold is increased, lower compression rates $\frac{|\mathcal{D}'|}{|\mathcal{D}|} \%$ are produced. Lower compression rates are accompanied by an increase in error as depicted by the box plots in Figure 6.7. As can be discerned from the images and CAD models, the relative positions of the curves correspond to the scene complexity: the more complex the scene, the higher the compression rate. Scenes (b) and (d) are the most complex, with noisy foliage, measurements of ceilings behind window panes and discontinuities as great as 5 m, compared to that of the 1 m discontinuity that the person in (c) presents. In all cases, these complex regions have been sampled most heavily – the outline of the person can be recognised in the Euclidean plot of (c) in Figure 6.4.

6.8.2 Surface Error

Box plots indicate surface error distributions for each of the test cases in Figure 6.7. The relative extent of errors between test cases are consistent with our scene assessment in Section 3.5 which used simple linear interpolation between all measurements. As compression rate decreases, measurement prediction errors typically migrate from being that of a tightly bunched distribution comparable to the measurement noise, to one with an increased mean and a significant tail – into the metre range in the worst case. A profile view of test case (a) is used to highlight how errors change with κ in Figure 6.8.

To demonstrate the spatial distribution of error we provide the surface predictions of test case (a) and (b) coloured according to error in Figure 6.9. For test case (b) we observe that the worst errors occur at discontinuities and for surfaces with high frequency components such as foliage. In these areas the performance is that of a naïve implementation that uses the entire dataset $\mathcal{D}$, as discussed in Section 6.4.

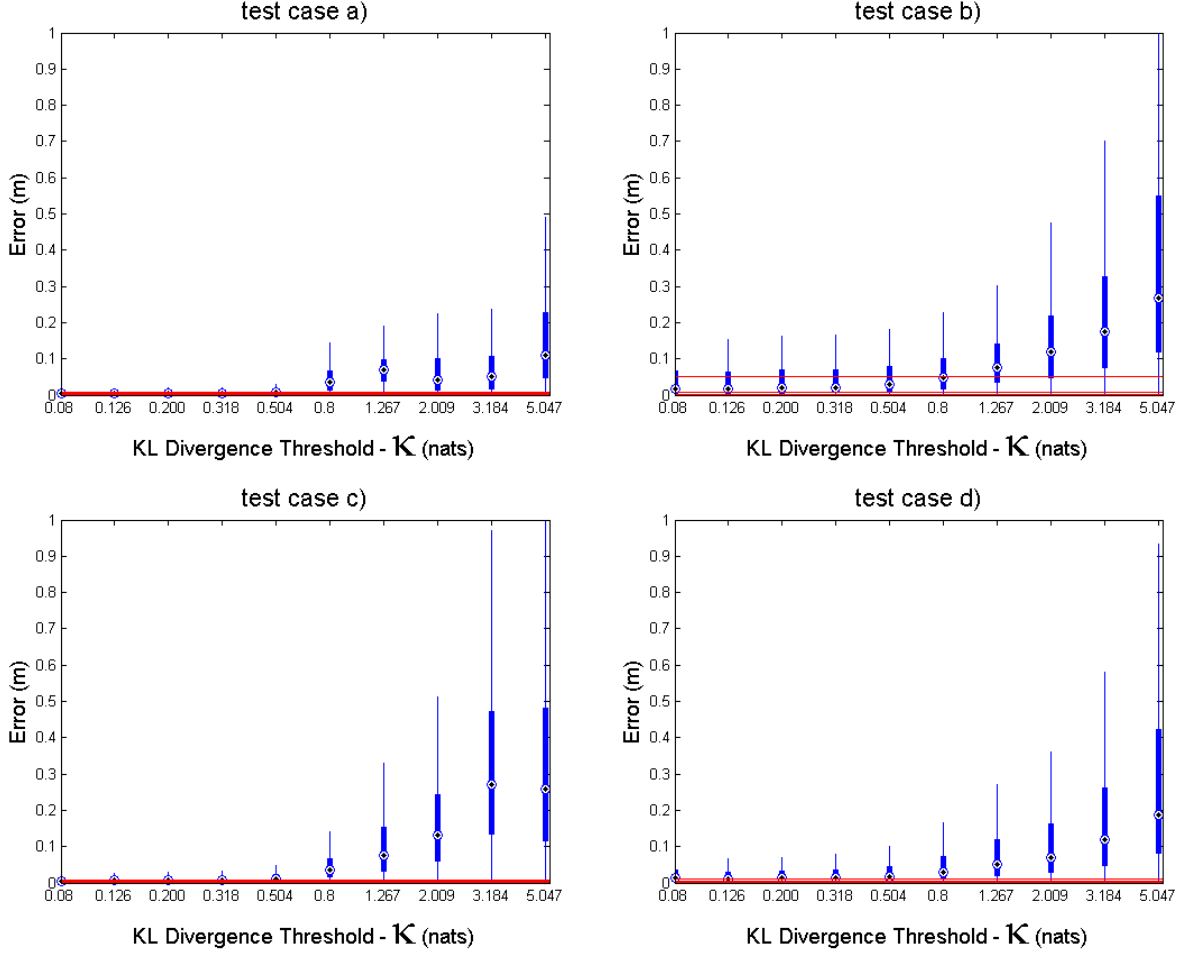


Figure 6.7: Error versus prediction threshold κ for test cases (a) to (d) as detailed in Section 6.7. Box plots denote median and interquartile range for each of the logarithmically spaced κ indicated by markers in Figure 6.6. Whiskers extend a maximum of one and a half times the interquartile range. Red lines indicate the median and interquartile range that result from using linear interpolation of all data points (as discussed in Section 3.5). They are provided to permit comparison to our approach. As κ increases both the width and the median of the distribution of the error increases. The extent of the migration follows roughly with the scene complexity, which can be seen in the images of Figure 6.4. As expected from our scene assessment in Section 3.5, scenes with a high complexity, such as test case (b) start with a significantly higher median error. We believe that particularly for test case (b) this is because of the extremely high frequency oscillations in range measurements $\mathbf{r}$ that are not fully captured by our predictive surfaces. These oscillations are due to, for example, alternating measurements from foreground leaves and background walls.

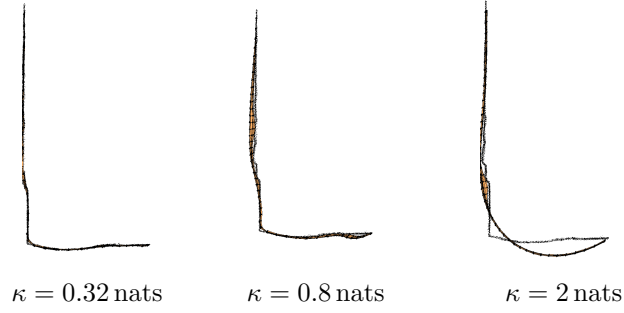


Figure 6.8: Profile view of a short section of wall and ground plane from test case (a) (Figure 6.4 and 6.9), with $\kappa = \{0.32, 0.8, 2\}$ nats from left to right. $\mathcal{D}$ is shown in black, while predictions conditioned on $\tilde{\mathcal{D}}$ are collated into the surfaces. Larger thresholds create small compression rates, hence smoother surfaces that are in greater error.

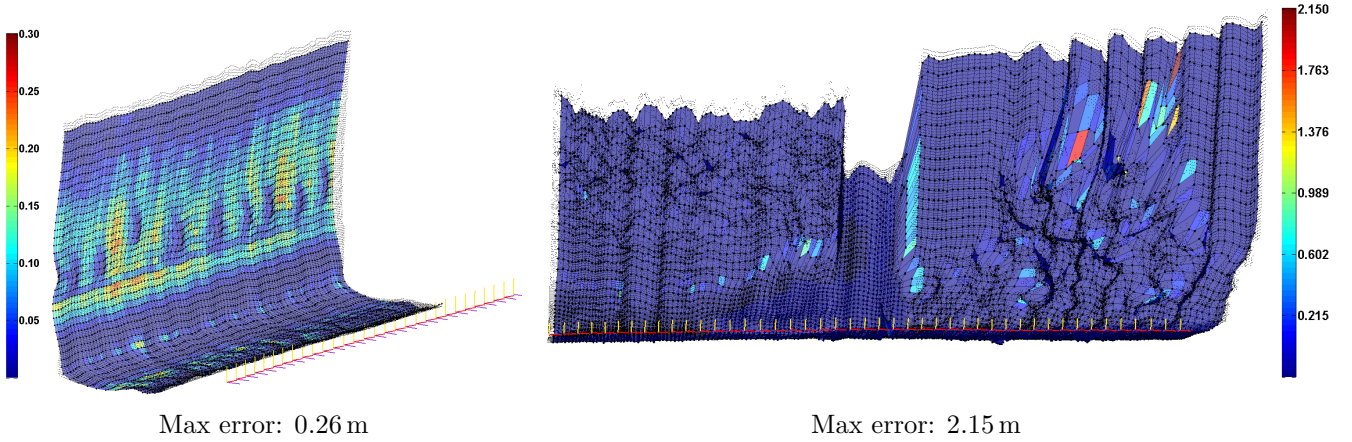


Figure 6.9: Error in surface predictions conditioned on the retained laser measurements $\tilde{\mathcal{D}}$ as the active window progressed across sensor configuration space $\mathcal{Q}$ (as per Section 6.3.1). Colouring is provided by the ℓ_1 norm in range from original measurements (in black) and the predicted surface, using $\kappa = 0.8$ nats. Colour scaling is different for each plot, but both use a *jet* colour map with high errors in red and low errors in blue. (Left) and (right) illustrate simple and complex regions from test case (a) and (b) from Figure 6.4. A profile view of a short section of (left) is also plotted in Figure 6.8. In (left), where there is an absence of large discontinuities, we notice the smallest errors are near $\mathcal{D}'$ (dark blue areas). In (right) there is a much more peaky distribution of error with significant errors only where there are large discontinuities and surfaces with high frequency components. In these regions there is automatically no data pruning and the accuracy of the surface is limited by that of a naïve implementation as discussed in Section 6.4.

6.8.3 Timing Considerations

We also analysed the time taken to compute our surface representation. Each of our examples (Figure 6.4) cover roughly 120,000 measurements of laser data of which we grid sample to use a quarter as input to our algorithm (see Chapter 3.5). In Figure 6.10 the solid blue line represents the time taken to process all laser measurements in each example for a given prediction threshold κ . The lower, dashed red line indicates when no introspection is permitted. The system was implemented with unoptimised MATLAB® code using one core of an AMD Opteron 8378 processor. While we ensured that updating the active window is order $\mathcal{O}(n^2)$ in the size of the active window, rather than that of $\mathcal{O}(n^3)$ of a naïve implementation, it is the most expensive operation in our algorithm. It is therefore an intuitive result that as κ increases the time taken decreases because the number of measurements used in our representation decreases. When introspection is permitted, there is a clear spike around $\kappa = 1$ nat. Here the algorithm stops accepting all measurements into the active window as the KL divergence between the measurement and its prediction falls below κ , hence introspection can begin to be applied. Introspection involves forming predictions of previously discarded measurements whenever new measurements are adopted into the active window. This potentially causes a large overhead in time. This behaviour could be mitigated somewhat by employing an intelligent introspection policy, whereby introspection terminates before reaching the back of the active window. Alternatively introspection could only be used around discontinuities, indicated by large KL divergences, and we return to this idea in Chapter 8.

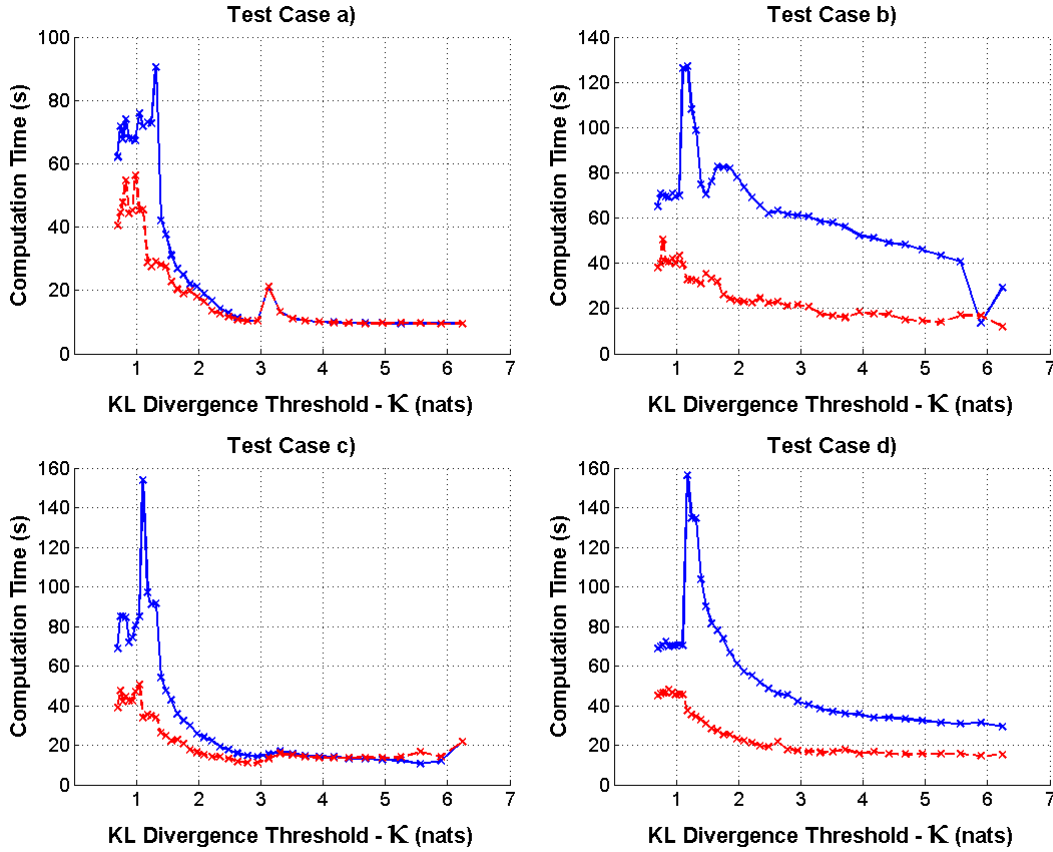


Figure 6.10: Computation time versus prediction threshold κ for each of the test cases (a) to (d) (see Figure 6.4) using unoptimised MATLAB® code. Reported times exclude time spent predicting hold out measurements (which are only used only for the production of error metrics) using introspection for the solid blue lines and without using introspection for the dashed red lines. Time taken follows roughly in accordance with scene complexity and decreases as κ increases. This is because the dominant computation in our algorithm is the updating of state on adoption of new laser measurements. This occurs more frequently with more complex scenes and lower κ . There is also a spike around $\kappa = 1$ nat for the blue line as around this value introspection begins to occur as not all laser measurements are adopted into the active subset.

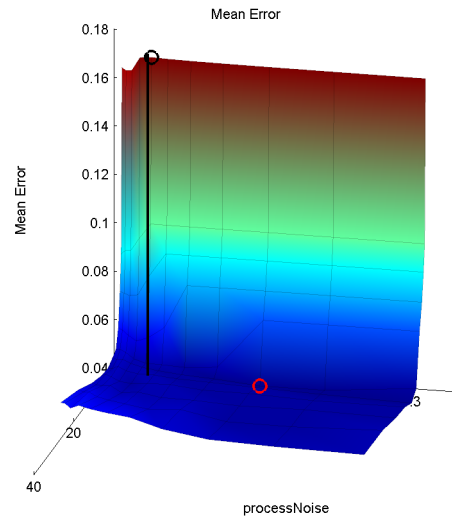
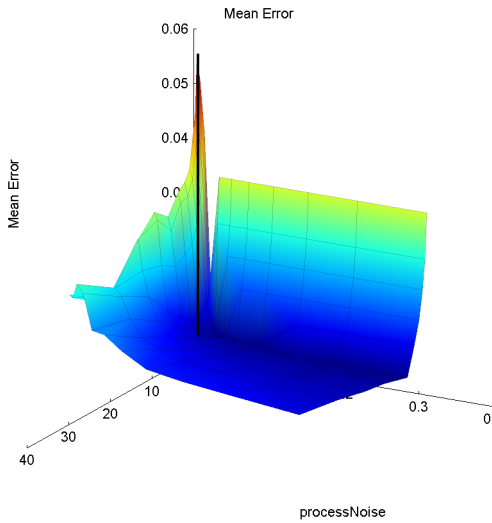
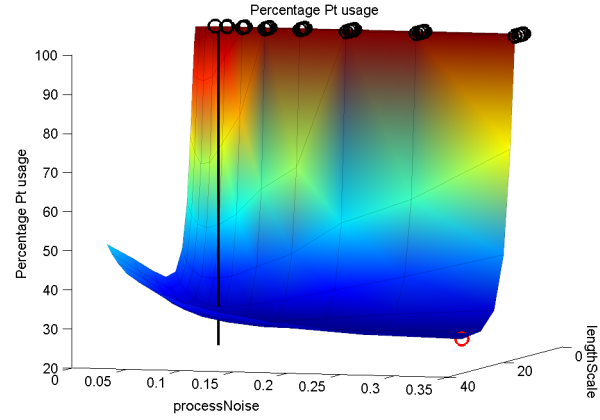
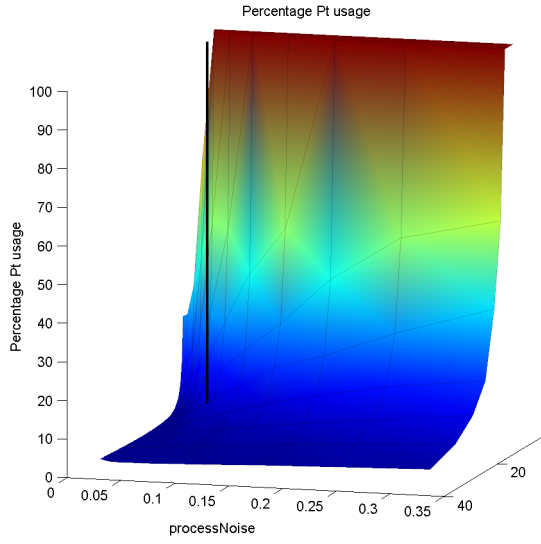
6.8.4 Choosing Hyperparameter Values

The hyperparameters ξ of our covariance function consist of length scale l and process variance σ_p^2 (see Section 5). To choose hyperparameter values we performed a sensitivity analysis over both length scale and process variance for a variety of prediction thresholds κ and different sections of the New College dataset.⁴ The cost function used for our optimisation aimed to minimise both mean error and compression rate (a visualisation of the cost surfaces for test cases (a) and (d) are included in Figure 6.11). This sensitivity analysis indicated that our algorithm is relatively insensitive to the choice of hyperparameter values, so long as both length scale and process noise were greater than roughly 4 units and 0.04 m^2 respectively. Mean error gradually increased past a length scale of roughly 6 units and a process variance of 0.3 m^2 . To allow comparison across all examples we chose length scale $l = 8$ units and process variance $\sigma_p^2 = 0.05 \text{ m}^2$ as they provided consistently accurate surface representations with aggressive compression rates. We return to the discussion of the variability between different workspace surfaces in the concluding section of this chapter and use it as motivation for extensions to our algorithm in Chapter 7.

6.8.5 Varying the Regression Model

We now turn our attention to the dependence of our results on the regression model which we use to form our predictions $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \xi) = \mathcal{N}(\mu_S, \sigma_S^2)$ conditioned on our active set of measurements $\tilde{\mathcal{D}}$. We use a GP prior with hyperparameters ξ and we

⁴Note that in Chapter 7 we provide an alternative approach whereby our model automatically weights relevant hyperparameter values highly for each prediction thus eschewing the need for this investigation.



test case (a)

test case (d)

Figure 6.11: Examples of the cost functions used in our sensitivity analysis for test case (a) and (d) from Chapter 3.5 (images repeated in Figure 6.4). All plots were for a prediction threshold $\kappa = 2.4$ nats. (Top) compression rate $\frac{|\mathcal{D}'|}{|\mathcal{D}|} \%$ and (bottom) mean error (in metres) for a variety of length scales l and process variances σ_p^2 . For all the sections of the New College dataset a longer length scale and a greater process variance produced both a low compression rate and a low mean error up until roughly 4 units and 0.04 m^2 respectively. Beyond these the gradient of the compression rate varied according to the overall workspace complexity of the scene, but was generally positive for mean error beyond roughly 6 units and 0.3 m^2 .

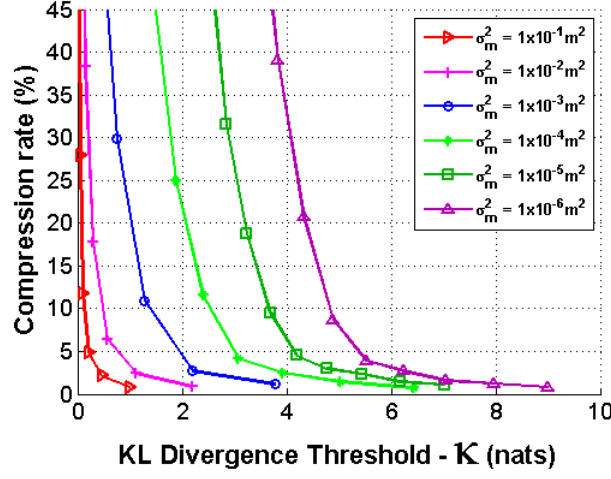


Figure 6.12: Compression rate versus κ for test case (c) (see Figure 6.4) for varying measurement variance σ_m^2 . Each line corresponds to a different measurement variance σ_m^2 , which is used in our measurement model to form our regression model (see Chapter 4). As measurement variance increases the curves move to the left. This is an intuitive result as the KL divergence between two distributions increases as the variance of each distribution decreases (see Equation 6.2). For comparison of associated error see box plots in Figures 6.7 and 6.13, where $\sigma_m^2 = 0.01 \text{ m}^2$ and $\sigma_m^2 = 1 \times 10^{-4} \text{ m}^2$ respectively. From Sections 7.2 onwards we use $\sigma_m^2 = 1 \times 10^{-4} \text{ m}^2$.

use additive Gaussian noise with zero mean and a variance of strength σ_m^2 for our measurement model (see Section 4.4). For a given κ one would expect that, as σ_m^2 decreases, the compression rate $\frac{|\mathcal{D}'|}{|\mathcal{D}|} \%$ would increase. This is because the KL divergence between two distributions is larger the smaller their variances (assuming their means are fixed and non-identical – see Equation 6.2), hence more measurements are adopted into the active set $\mathcal{D}'$. This is demonstrated in Figure 6.12 by the migration of the curves to the right as σ_m^2 decreases. We include Figure 6.13 with $\sigma_m^2 = 1 \times 10^{-4} \text{ m}^2$ rather than $\sigma_m^2 = 1 \times 10^{-2} \text{ m}^2$ that is used in previous sections, to allow comparison of the resulting errors.

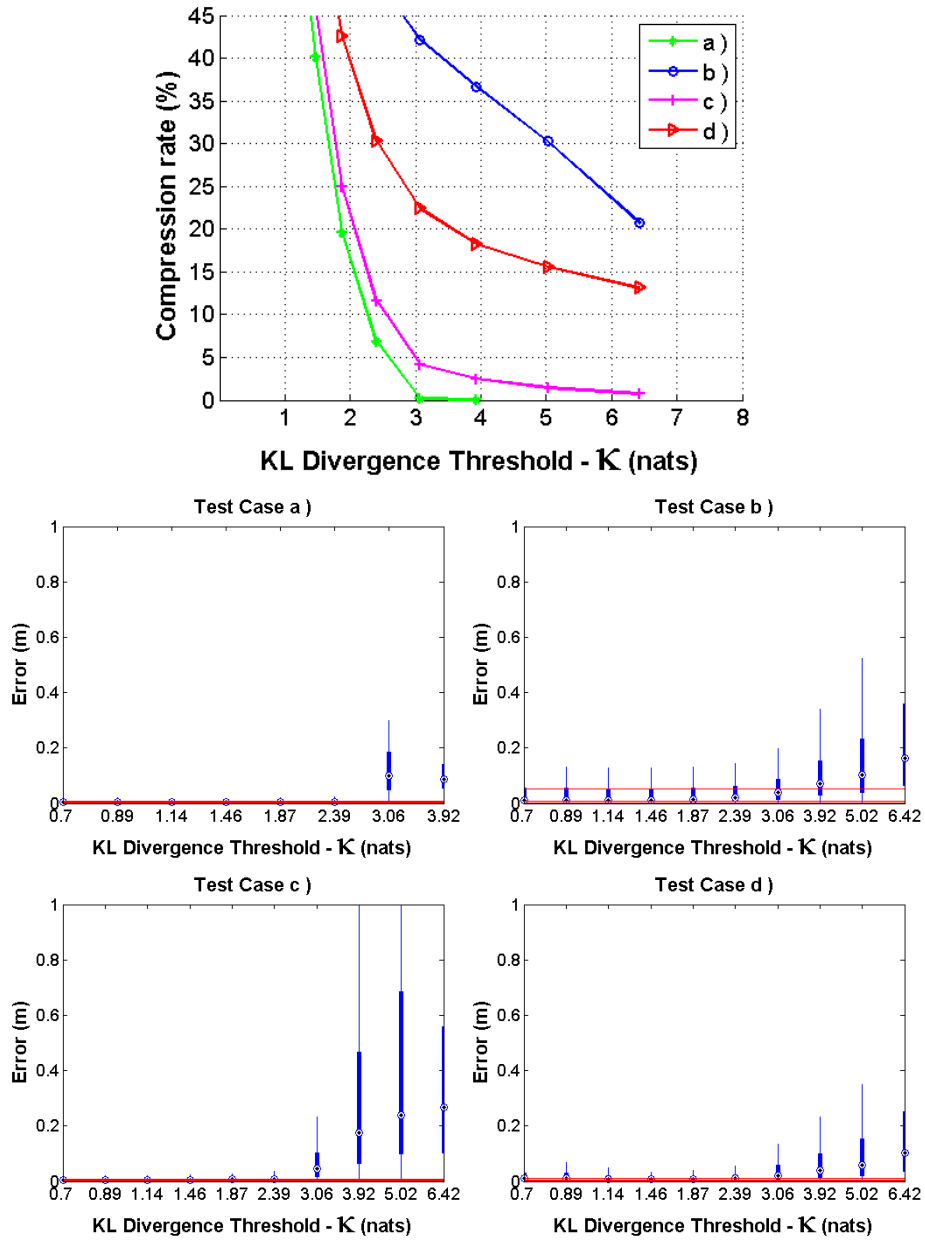


Figure 6.13: Summary plots of the test cases in Figure 6.4 demonstrating the effect of varying κ for a measurement variance $\sigma_m^2 = 1 \times 10^{-4} \text{ m}^2$. These plots are provided for comparison to Figures 6.6 and 6.7 where $\sigma_m^2 = 0.01 \text{ m}^2$. Red lines also indicate the median and interquartile range that result from using linear interpolation of all data points (as discussed in Section 3.5). (Left) compression rate $\frac{|\mathcal{D}'|}{|\mathcal{D}|} \%$ versus κ . As prediction threshold κ increases, compression rate decreases. By comparing with Figure 6.6 we note that when using a greater measurement variance for test case (b) (Figure 6.6) a lower compression rate is produced relative to the other test cases. There is a significant quantity of foliage in test case (b), which produces high frequency oscillations in $\mathbf{r}$ as measurements in these regions tend to alternate between leaves and the walls behind them. A high σ_m^2 will produce a smoother surface (see Section 4.4.1.2), which may on average fit training data more effectively and hence achieve a lower compression rate in Figure 6.6. Box plots (right) denote median and interquartile range for each of the κ indicated by markers in the plot on the left. Whiskers extend a maximum of one and a half times the interquartile range.

6.9 Conclusions

In this chapter we have adopted an online formulation of Gaussian Process Regression to regress laser range measurements in Sensor Configuration space. Our framework provides a continuous representation of the implicit workspace surfaces underlying the data while at the same time enables a principled subsampling of the point-cloud via an information theoretic data selection criterion. The resulting algorithm decimates point clouds of simple workspaces by factors in excess of two orders of magnitude without significant degradation in fidelity. The computational complexity of the algorithm proposed is reduced by adopting a sliding window approach and by managing variables that are used throughout the active sampling process. The final complexity is quadratic in the size of the active window, which we engineer to be constant, rather than cubic in the size of the dataset.

Throughout, we have used a stationary GP with a parametric mean function and the Matérn class of covariance function, although we noted that many others could be substituted in their place. A sensitivity analysis over our hyperparameters revealed that our algorithm is relatively insensitive to the choice of hyperparameter values which was manifested by large plateau regions in our error metrics. We set values according to a rough sweet spot that we identified within these regions which was consistent across the New College dataset. This sweet spot reflects the overall natural workspace properties that are present in our laser data. However, we found that the distribution of surface error was sometimes non-uniform with greater errors coinciding with areas of sensor configuration space that have highly complex surfaces as we demonstrated in Figure 6.9b. We therefore posit that although intrinsic error is high in these regions (due

to our method of error assessment - see Chapter 3.5), that improvements in surface accuracy can come from a model with greater flexibility. In Chapter 7 we achieve this through lifting the restriction of a single fixed set of hyperparameter values and instead effectively allow them to vary across sensor configuration space in response to the *local* workspace conditions.

Chapter 7

Non-Stationarity in an Active Window Framework

The non-stationary active window approach that is described in this chapter was published as ‘Adaptive Compression For 3D Laser Data’ in the International Journal of Robotics Research, 2011.

We now provide refinements to the stationary GP framework that we presented in Chapter 6. Whilst our approach produced aggressive compression rates for a relatively small loss in accuracy, we are motivated by the desire to further increase surface prediction accuracy for an even lower set of retained measurements. We achieve this through endowing our algorithm the ability to form predictions over a range of GP models and by weighting each according to changing workspace conditions. Our windowed framework naturally lends itself to this task and although we continue to use the stationary GP that was introduced in Chapter 6 we achieve a framework that is overall

non-stationary.

To achieve our non-stationary model we extend our Bayesian treatment of unknown variables (see Chapter 4) through the marginalisation of our model hyperparameters in addition to our model parameters. To approximate this intractable integral we turn to Bayesian Quadrature (BQ) [73, 89] (a form of probabilistic numerical integration) which has been successfully applied to the marginalisation of GP hyperparameters for 1D regression in [77]. As stepping stones towards this goal we investigate two alternative approximations to BQ, namely a simple Monte Carlo method and a type-II Maximum Likelihood solution. Both remove the prohibitive computational cost of the full approach whilst retaining a coarse degree of automatic control of the hyperparameters. We use the results of our findings to discuss the design of a full Bayesian treatment as future work.

7.1 New Nomenclature Introduced in this Chapter

Non-stationary Gaussian Process Regression

- $\cdot^*$ - subset of a GP concerning input space $\mathcal{E}$
- g - number of hyperparameter samples ξ_i
- $\mathcal{E}$ - hyperparameter sample space
- ξ_i - hyperparameter sample $\xi_i \in \mathcal{E}$
- Ξ - set of all hyperparameter samples $\Xi = \{\xi_i\}_{i=1}^g$
- $\phi(\xi)$ - GP Prior over marginal likelihood $p(\mathbf{r}|Q, \xi)$
- $v(\xi)$ - GP Prior over posterior predictive distribution $p(r^*|\mathbf{q}^*, \mathcal{D}, \xi)$
- $p(\xi)$ - prior over hyperparameter space $\mathcal{E}$
- $p(r^*|\mathbf{q}^*, \mathcal{D})$ - marginalised posterior predictive distribution
- $p(r^*|\mathbf{q}^*, \mathcal{D}, \Xi)$ - approximation of marginalised posterior distribution as given
by Bayesian Quadrature approach
- μ_{GMM} - mean of our Gaussian Mixture Model (GMM) approximation
- σ_{GMM}^2 - variance of our Gaussian Mixture Model (GMM) approximation
- ρ - mixing coefficients (posterior predictive distribution weightings)
- ξ_{ML-II} - single hyperparameter sample with maximum marginal likelihood
($\max_i p(\mathbf{r}|Q, \xi_i)$) as used in our simple approximation of ρ
- $p(\xi_i|\mathcal{D})$ - posterior of the marginal likelihood under a uniform prior
($\rho_i = p(\xi_i|\mathcal{D})$) as used in our approximation of ρ

7.2 Non-Stationarity through a Data Dependent Gaussian Mixture Model

Our aim is to allow our Gaussian Process Regression (GPR) algorithm to exploit models that perform well for rough surfaces, in rough areas of the workspaces (such as foliage); and models aimed at smooth surfaces, in smoother areas of the workspace (such as walls). Using the notation from Chapter 6, consider using a larger length scale l for our GP covariance function in a region of sensor configuration space $\mathcal{Q}$. In this region each laser measurement will influence predictions over large distances in $\mathcal{Q}$. A relatively smooth predictive surface ensues (see Section 4.4.1.2) that is more suited to regions of low complexity than those surfaces generated from shorter length scales. The accuracy of our predictions will therefore increase. As each measurement has more affect on predictions, fewer measurements are required per unit hypervolume of $\mathcal{Q}$ to support predictions. Compression rate in these regions will therefore also decrease. Computational requirements are also reduced due to the increase in the span of the active window η which permits the same window to be used for a greater number of predictions.

The computational savings that we presented in Chapter 6 (where we achieved an $\mathcal{O}(n^2)$ algorithm rather than $\mathcal{O}(n^3)$ of a naïve application of GPR, where n is the number of measurements in the active set) are dependent on using a stationary GP covariance function (and thus our stationary regression model) that is fixed across the entire workspace. We therefore form our measurement predictions from a weighted linear sum of estimates from a bank of fixed GP regressors, each maintained as per

Chapter 6. By allowing the weighting to vary in response to the current workspace conditions we achieve a regressor that is non-stationary overall. This Gaussian Mixture Model (GMM) approach is a natural result of Bayesian Quadrature (as we shall see in Section 7.3.2), scales linearly with the number of GP instantiations and is trivial to parallelise.

The Marginal Likelihood of measurements given each of our stationary GP models provide a convenient means to weight our GMM model based on the current set of measurements in our active window $\tilde{\mathcal{D}}$. We use the Marginal Likelihood instead of relying on expert knowledge as in [85]. It appears as a consequence of the Bayesian treatment of hyperparameters as we shall see in Section 7.3 and we shall discuss its calculation in Section 7.3.1. We then briefly overview the Bayesian Quadrature approach in Section 7.3.2 and its application to Gaussian Process Regression in Section 7.3.3. To exploit the increased model flexibility that this approach affords we make simple modifications to our stationary covariance function (presented in Section 5.2) in Section 7.3.4. Our approximations are then introduced in Section 7.4 and an in depth comparison between each approximation and our stationary GP is presented in Section 7.5. Finally we discuss the considerations that are required to apply the full Bayesian Quadrature approach in practice in Section 7.6.

7.3 Bayesian Approach to Non-Stationarity

In Chapter 4 we formed our predictions $r^\star$ through marginalising our model parameters ζ and conditioning on training data $\tilde{\mathcal{D}} = \{(\tilde{\mathbf{q}}_i, \tilde{r}_i)\}_{i=1}^n$ where n is the size of the active

set (and not the entire dataset N) and $\mathbf{q} \in \mathcal{Q}$ is a sensor configuration in sensor configuration space

$$\begin{aligned} p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi}) &= \int p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\zeta}, \boldsymbol{\xi}) p(\boldsymbol{\zeta}|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi}) d\boldsymbol{\zeta} \\ &\approx \int p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\zeta}, \boldsymbol{\xi}) p(\boldsymbol{\zeta}|\mathbf{q}^*, \tilde{Q}, \boldsymbol{\xi}) d\boldsymbol{\zeta} \end{aligned} \quad (7.1)$$

We assumed a likelihood model (Equation 4.40 conditioned on $\tilde{\mathcal{D}}$) $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\zeta}, \boldsymbol{\xi})$ with additive Gaussian noise to account for sensor inaccuracies. Our GP prior $p(\boldsymbol{\zeta}|\mathbf{q}^*, \tilde{Q}, \boldsymbol{\xi})$ (Equation 4.35) described a distribution over latent functions underlying our observation model. We were left with a posterior predictive distribution $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi})$ (Equation 4.46) which was dependent on our hyperparameters to specify the general shape of our regression.

We now also aim to model average over our hyperparameters $\boldsymbol{\xi}$ through a second level of marginalisation in addition to the marginalisation of model parameters $\boldsymbol{\zeta}$ in Equation 7.1. To achieve this marginalised posterior predictive distribution $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}})$ we rearrange the joint probability of the predictions r^* , data $\tilde{\mathcal{D}}$ and hyperparameters $\boldsymbol{\xi}$

$$\begin{aligned} p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}) &= \frac{\int p(r^*, \tilde{\mathbf{r}}, \boldsymbol{\xi}|\mathbf{q}^*, \tilde{Q}) d\boldsymbol{\xi}}{p(\tilde{\mathbf{r}}|\mathbf{q}^*, \tilde{Q})} \\ &= \frac{\int p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi}) p(\tilde{\mathbf{r}}|\mathbf{q}^*, \boldsymbol{\xi}, \tilde{Q}) p(\boldsymbol{\xi}|\mathbf{q}^*, \tilde{Q}) d\boldsymbol{\xi}}{\int p(\tilde{\mathbf{r}}|\mathbf{q}^*, \boldsymbol{\xi}, \tilde{Q}) p(\boldsymbol{\xi}|\mathbf{q}^*, \tilde{Q}) d\boldsymbol{\xi}} \\ &\approx \frac{\int p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi}) p(\tilde{\mathbf{r}}|\tilde{Q}, \boldsymbol{\xi}) p(\boldsymbol{\xi}) d\boldsymbol{\xi}}{\int p(\tilde{\mathbf{r}}|\tilde{Q}, \boldsymbol{\xi}) p(\boldsymbol{\xi}) d\boldsymbol{\xi}} \end{aligned} \quad (7.2)$$

where

$$\tilde{\mathcal{D}} = \{(\tilde{\mathbf{q}}_i, \tilde{r}_i)\}_{i=1}^n \quad (7.3)$$

$$p(\tilde{\mathbf{r}}|\mathbf{q}^*, \tilde{Q}, \boldsymbol{\xi}) = p(\tilde{\mathbf{r}}|\tilde{Q}, \boldsymbol{\xi}) \quad (7.4)$$

$$p(\boldsymbol{\xi}|\mathbf{q}^*, \tilde{Q}) = p(\boldsymbol{\xi}) \quad (7.5)$$

where $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi})$ is our stationary GP prediction given in Equation 7.1. The Marginal Likelihood of the measurements given hyperparameters $\boldsymbol{\xi}$, $p(\tilde{\mathbf{r}}|\tilde{Q}, \boldsymbol{\xi})$ can be readily calculated for a parametric mean GP as we shall report in Section 7.3.1.¹ Like all Bayesian techniques, we explicitly account for workspace knowledge in the prior $p(\boldsymbol{\xi})$. Unlike in Equation 7.1, we do not have an analytic solution to either of the marginalisations in Equation 7.2. Instead we turn to quadrature techniques and approximate the integrals by drawing *hyperparameter samples* $\boldsymbol{\xi}_i$ (where $\Xi = \{\boldsymbol{\xi}_i\}_{i=1}^g$ is the set of all hyperparameter samples of size g) from each of the distributions over hyperparameters: $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi})$ and $p(\tilde{\mathbf{r}}|\tilde{Q}, \boldsymbol{\xi})$. We arrive at an approximate marginalised posterior distribution $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \Xi) \approx p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}})$.

¹The Marginal Likelihood for a deterministic mean GP is given in Equation 4.42.

7.3.1 Log Marginal Likelihood of Parametric Mean GPs

The calculation of the log Marginal Likelihood for the parametric mean GP $S(\cdot)$ presented in Chapter 6 is standard as reported by [88]²

$$\begin{aligned} \log p(\tilde{\mathbf{r}}|\tilde{Q}, \boldsymbol{\xi})_S = & -\frac{1}{2} \left[\mathbf{r}^T K_{\tilde{\mathbf{r}}}^{-1} \mathbf{r} - (n-m) \log 2\pi \dots \right. \\ & \left. - \log |K_{\tilde{\mathbf{r}}}| - \log |A| + W \right] \end{aligned} \quad (7.6)$$

where

$$W = \mathbf{r}^T K_{\tilde{\mathbf{r}}}^{-1} H^T A^{-1} H K_{\tilde{\mathbf{r}}}^{-1} \mathbf{r} \quad (7.7)$$

and m is the rank of H . All elements follow the same definitions as previously defined in Chapters 4 and 6. The account of Equation 7.6 in [88] refers to a paper of considerable length [7], of which we have simply assumed true for the purposes of our work.

In Section 4.4.2 we showed an efficient technique to calculate $\log |K_{\tilde{\mathbf{r}}}|$ (which may be incrementally updated following [77]), however the complexity of Equation 7.6 is dominated by the calculation of the rank m of H . Off-the-shelf implementations based on Singular Value Decomposition such as [61] can estimate m in $\mathcal{O}(n^2)$ (where n is the size of the active set). These implementations have the same complexity as that of updating our stationary GP model hence there is no adverse effect on the complexity of our framework.

²The log Marginal Likelihood for a deterministic mean GP is given in Equation 4.43.

7.3.2 Simple Bayesian Quadrature

To approximate the integrals in Equation 7.2 we extend our Bayesian approach through the use of Bayesian Quadrature (sometimes called Bayesian Monte Carlo or Bayes-Hermite Quadrature) [73, 89, 77]. This technique provides a sampling method that, unlike classical Monte Carlo [57], accounts for both the sample values and their positions through the use of uncertainty as advocated in [72]. In our treatment we shall consider only those equations pertinent to the understanding of Bayesian Quadrature in preparation for the approximations that we make later in Section 7.4 (for a full discussion see [77]).³

Bayesian Quadrature is a means to perform approximate Bayesian inference of the potentially non-analytic marginalisation of an unknown function $f(\boldsymbol{\xi})$ with respect to variable $\boldsymbol{\xi} \in \mathcal{E}$ (where $\mathcal{E}$ is hyperparameter samples space) with prior distribution $p(\boldsymbol{\xi})$

$$\pi \triangleq \int f(\boldsymbol{\xi})p(\boldsymbol{\xi}) d\boldsymbol{\xi} \quad (7.8)$$

The evaluation of $f(\boldsymbol{\xi})$ at all possible locations in $\mathcal{E}$ (which we shall denote as the infinite length vector $\mathbf{f}_o \triangleq \mathbf{f}(\boldsymbol{\xi}_o)$ - see Section 4.4) is generally intractable. Instead, Bayesian Quadrature requires that we are able to evaluate $f(\boldsymbol{\xi})$ at sample locations $\mathbf{f} \triangleq \mathbf{f}(\Xi)$. To permit this we form an estimate of the integral in Equation 7.8 $p(\pi|\mathbf{f}, \Xi)$ through the use of a GP to represent $f(\boldsymbol{\xi})$

³For clarity we use $.^*$ to indicate the subset of a GP concerning sensor configuration space $\mathcal{Q}$ that we use for predictions (as we have done in previous chapters) and $.^o$ to indicate the subset of a GP concerning hyperparameter sample space $\mathcal{E}$ that we use for predictions. Where we have used $.o$ in this chapter it is always in relation to a GP concerning $\mathcal{E}$.

$$f(\boldsymbol{\xi}) \sim \mathcal{GP}(\mathbf{0}, k(\boldsymbol{\xi}, \boldsymbol{\xi}')) \quad (7.9)$$

This GP is used to provide us with a prediction $p(f^*|\boldsymbol{\xi}^*, \mathbf{f}, \Xi)$ of the underlying function $f(\boldsymbol{\xi})$ at location $\boldsymbol{\xi}^*$ which is conditioned on the finite number of observations $\mathbf{f}$ at locations Ξ . In the following we shall show that Equation 7.8 becomes tractable under this approximation.

As discussed in Chapter 4, a GP represents a distribution over functions such as $\mathbf{f}_o$, of which we may form predictions $p(\mathbf{f}_o|\mathbf{f}, \Xi)$ conditioned on our observations $\mathbf{f}$ at locations Ξ . To evaluate our approximation $p(\pi|\mathbf{f}, \Xi)$ we follow the Bayesian treatment of unknown variables which is to marginalise $\mathbf{f}_o$ (using the machinery already introduced in Chapter 4 and Appendix D)

$$p(\pi|\mathbf{f}, \Xi) = \int p(\pi|\mathbf{f}_o, \mathbf{f}, \Xi) p(\mathbf{f}_o|\mathbf{f}, \Xi) d\mathbf{f}_o \quad (7.10)$$

where the likelihood $p(\pi|\mathbf{f}_o, \mathbf{f}, \Xi)$ is

$$p(\pi|\mathbf{f}_o, \mathbf{f}, \Xi) = \delta\left(\pi(f) - \int f(\boldsymbol{\xi}) p(\boldsymbol{\xi}) d\boldsymbol{\xi}\right) \quad (7.11)$$

$$= \lim_{\sigma_\pi^2 \rightarrow 0} \mathcal{N}\left(\pi(f); \int f(\boldsymbol{\xi}) p(\boldsymbol{\xi}) d\boldsymbol{\xi}, \sigma_\pi^2\right) \quad (7.12)$$

$\delta(\pi - a)$ is the Dirac delta density in π centred at a . It has been used to indicate that π is deterministic on the unknown function $f(\boldsymbol{\xi})$ (as given by Equation 7.8) and

it has the effect of selecting only the true underlying function $\mathbf{f}_o$ in the marginalisation of Equation 7.10.

By re-writing Equation 7.11 as a normal distribution with variance σ_π^2 that tends to zero in the limit and by noting that integration is a linear operator (see [77] for a discussion), $p(\pi|\mathbf{f}_o, \mathbf{f}, \Xi)$ and $p(\mathbf{f}_o|\mathbf{f}, \Xi)$ therefore form a linear Gaussian model (see Section 4.2.1). We may hence use Equation 4.14 to perform the marginalisation of Equation 7.10 with result

$$p(\pi|\mathbf{f}, \Xi) = \mathcal{N}(\pi; \mathbb{E}[\pi|\mathbf{f}, \Xi], \mathbb{V}[\pi|\mathbf{f}, \Xi]) \quad (7.13)$$

where

$$\begin{aligned} \mathbb{E}[\pi|\mathbf{f}, \Xi] &= \int p(\boldsymbol{\xi}^*) \mathbb{E}[f^*|\boldsymbol{\xi}^*, \mathbf{f}, \Xi] d\boldsymbol{\xi}^* \\ &= \left(\int p(\boldsymbol{\xi}^*) \mathbf{k}(\Xi, \boldsymbol{\xi}^*)^T d\boldsymbol{\xi}^* \right) K_{\mathbf{f}, \mathcal{E}}^{-1} \mathbf{f} \\ &= \chi^T K_{\mathbf{f}, \mathcal{E}}^{-1} \mathbf{f} \end{aligned} \quad (7.14)$$

$$\begin{aligned} \mathbb{V}[\pi|\mathbf{f}, \Xi] &= \iint p(\boldsymbol{\xi}^*) \mathbb{V}[f|\boldsymbol{\xi}^*, \mathbf{f}, \Xi] p(\boldsymbol{\xi}^{*'}) d\boldsymbol{\xi}^* d\boldsymbol{\xi}^{*'} \\ &= \iint p(\boldsymbol{\xi}^*) k(\boldsymbol{\xi}^*, \boldsymbol{\xi}^{*'}) p(\boldsymbol{\xi}^{*'}) d\boldsymbol{\xi}^* d\boldsymbol{\xi}^{*'} \dots \\ &\quad - \iint p(\boldsymbol{\xi}^*) \mathbf{k}(\Xi, \boldsymbol{\xi}^*)^T K_{\mathbf{f}, \mathcal{E}}^{-1} \mathbf{k}(\Xi, \boldsymbol{\xi}^{*'}) p(\boldsymbol{\xi}^{*'}) d\boldsymbol{\xi}^* d\boldsymbol{\xi}^{*'} \\ &= \gamma - \chi^T K_{\mathbf{f}, \mathcal{E}}^{-1} \chi \end{aligned} \quad (7.15)$$

We have used the subscript on the covariance matrix $K_{\mathbf{f}, \mathcal{E}}^{-1}$ to indicate that it is constructed from noiseless observations $\mathbf{f}$ of our underlying function $f(\boldsymbol{\xi})$ at hyperpa-

parameter sample locations $\boldsymbol{\xi}_i \in \mathcal{E}$ (not sensor configurations $\mathbf{q} \in \mathcal{Q}$). Therefore mean $\mathbb{E}[f^*|\boldsymbol{\xi}^*, \mathbf{f}, \Xi]$ and variance $\mathbb{V}[f^*|\boldsymbol{\xi}^*, \mathbf{f}, \Xi]$ of $p(f^*|\boldsymbol{\xi}^*, \mathbf{f}, \Xi)$ are those of the zero mean GP representing $f(\boldsymbol{\xi})$ that we defined in Equation 7.9. $\mathbb{E}[f^*|\boldsymbol{\xi}^*, \mathbf{f}, \Xi]$ and $\mathbb{V}[f^*|\boldsymbol{\xi}^*, \mathbf{f}, \Xi]$ are given in Equation 4.46 using a likelihood model with zero measurement noise σ_m^2 (see Equations 4.40 and 4.44).

We have grouped the marginalisations in Equations 7.14 and 7.15 with respect to $\boldsymbol{\xi}^*$ into χ and γ , which are in general non-tractable. For a conjugate normal distribution $p(\boldsymbol{\xi}) = \mathcal{N}(\boldsymbol{\xi}; \boldsymbol{\mu}, \Sigma)$ however there is an analytic solution (see [77]) which exploits the property that the integral of a product of two Gaussians is another (un-normalised) Gaussian. There are also a number of derivations for alternative distributions presented in [77]. As we shall see in Sections 7.4.2 and 7.4.3 we avoid the explicit computation of these approximations so we do not consider them further.

We are interested in the case that $f(\boldsymbol{\xi})$ is a likelihood $p(y|\boldsymbol{\xi})$ and in particular the posterior predictive distribution $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi})$ of our stationary GP (Equation 7.1) which was used in the marginalisation of our hyperparameters in Equation 7.2.⁴ Equation 7.8 becomes

$$\pi \triangleq p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}) = \int p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi}) p(\boldsymbol{\xi}) d\boldsymbol{\xi} \quad (7.16)$$

We form an estimate of this integral $p(\pi|\mathbf{q}^*, \tilde{\mathcal{D}}, \Xi)$ through the use a GP to approximate $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi})$. Specifically, the GP permits us to form predictions $p(r^{**}|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi}^*, \Xi)$

⁴Strictly any likelihood must be non-negative at all locations in $\mathcal{E}$. There is ongoing work by the author of [77] to ensure that this always the case when using a GP approximation. In practice this is only a problem when there is insufficient support from $\mathbf{f}$ which can be ameliorated through taking more observations (see Section 7.6 for further discussion on conditioning).

of the underlying distribution $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi})$ at locations $\boldsymbol{\xi}^*$ conditioned on the finite number of observations $\mathbf{r}$ at locations Ξ in $\mathcal{E}$.

Note that π is a distribution in Equation 7.16 and therefore $p(\pi|\mathbf{q}^*, \tilde{\mathcal{D}}, \Xi)$ is a distribution over a distribution (sometimes termed a second order probability). In order to recover our desired prediction $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \Xi)$ we must therefore marginalise π

$$\begin{aligned} p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \Xi) &= \int p(r^*|\pi, \mathbf{q}^*, \tilde{\mathcal{D}}, \Xi) p(\pi|\mathbf{q}^*, \tilde{\mathcal{D}}, \Xi) d\pi \\ &= \int \pi p(\pi|\mathbf{q}^*, \tilde{\mathcal{D}}, \Xi) d\pi \\ &= \mathbb{E}[\pi|\mathbf{q}^*, \tilde{\mathcal{D}}, \Xi] \end{aligned} \tag{7.17}$$

Hence when making predictions $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \Xi)$ we are only ever interested in $\mathbb{E}[\pi|\mathbf{q}^*, \tilde{\mathcal{D}}, \Xi]$ as given in Equation 7.14 (where r^* is substituted for f) and not the complete distribution of Equation 7.13. Re-writing Equation 7.14 we also note that predictions are simply a linear combination of samples

$$p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \Xi) = \sum_{i=1}^g \rho_i p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi}_i) \tag{7.18}$$

where the mixing coefficients $\boldsymbol{\rho}$ (otherwise known as weightings) are constrained by $\sum_{i=1}^g \rho_i = 1$ and are given by

$$\boldsymbol{\rho} = \chi^T K_{\tilde{\mathbf{r}}, \mathcal{E}}^{-1} \tag{7.19}$$

This is an intuitive result that we have already encountered in Chapter 4 (see Equa-

tion 4.33 and 4.49), save here our marginalisation is over model hyperparameters ξ instead of parameters ζ and hence we produce a linear combination of the stationary GP predictions $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \xi_i)$ (from Chapter 4) instead of the training measurements $\tilde{\mathbf{r}}$. Overall we therefore have a hierarchical regression model and since $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \xi_i)$ are Gaussian distributed it is also a Gaussian Mixture Model (GMM).

Importantly, Bayesian Quadrature has three properties expedient for our windowed regression approach. The first is that ρ is purely a function of the locations of our hyperparameter samples ξ_i in input space $\mathcal{E}$. If we fix these locations (which is required in order to exploit our efficient implementation as discussed in Section 7.2) then we need only evaluate ρ once before any measurements are considered, thus reducing our computation requirements at run-time. The second is that although we make observations of our unknown distribution $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \xi_i)$ at a limited number of locations Ξ , Bayesian Quadrature attempts to make the best use of these samples through approximating this distribution (through χ and $K_{\tilde{\mathbf{r}}, \mathcal{E}}$), whereas Monte Carlo techniques do not (see our approximation in Section 7.4.2). Of practical significance, we may also explicitly incorporate prior knowledge $p(\xi)$ which, as we shall see in Section 7.4.1, is important in order to avoid overfitting.

7.3.3 Bayesian Quadrature for Gaussian Process Regression

Osborne [77] notes that the correlations induced by $p(\tilde{\mathbf{r}}|\tilde{Q}, \xi)$ appearing in both the numerator and the denominator of the marginalisation of ξ in Equation 7.2 imply that Bayesian Quadrature must be performed on the equation as a whole. Two GP priors $v(\xi)$ and $\phi(\xi)$ are used to approximate the posterior predictive distribution

$p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi})$ and the Marginal Likelihood $p(\tilde{\mathbf{r}}|\tilde{\mathcal{Q}}, \boldsymbol{\xi})$ respectively, following the same technique that was described in Section 7.3.2. The overall technique is called *Predictive Bayesian Quadrature* of which Gaussian priors $p(\boldsymbol{\xi}) \sim \mathcal{N}(\boldsymbol{\xi}; \boldsymbol{\mu}, \Sigma)$ were mainly used [77] although discussion and derivations for other priors were also provided. The required marginalisation when using these two priors is non-analytic, hence a number of approximations are suggested in [77], each of which require different amounts of additional computation. Of these, the Maximum A Posteriori (MAP) approximation of the GP prior that approximates the Marginal Likelihood $\phi_{MAP}(\Xi) = \lim_{K_{\phi, \varepsilon} \rightarrow 0} \mathcal{N}(\phi(\Xi); \boldsymbol{\mu}_{\phi}(\Xi), K_{\phi, \varepsilon})$ preserves each of the three desirable properties of Bayesian Quadrature for our windowed approach (see Section 7.3.2). It is akin to assuming that we know the distribution over the Marginal Likelihood $p(\phi^*|\boldsymbol{\xi}^*, \boldsymbol{\phi}, \Xi)$ with infinite precision. The overall marginalised predictive distribution $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \Xi)$ is again a GMM using the posterior distribution of each of our stationary GPs $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi}_i) = \mathcal{N}(r^*; \mu_{S_i}, \sigma_{S_i})$, combined via mixing coefficients $\boldsymbol{\rho}$

$$p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \Xi) = \sum_{i=1}^g \rho_i \mathcal{N}(r^*; \mu_{S_i}, \sigma_{S_i}) \quad (7.20)$$

The mean μ_{GMM} and variance σ_{GMM}^2 of a summation of conditionally independent Gaussians are easily extracted:

$$p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \Xi) \approx \mathcal{N}(r^*; \mu_{GMM}, \Sigma_{GMM}) \quad (7.21)$$

where

$$\mu_{GMM} = \boldsymbol{\rho}^T \boldsymbol{\mu}_S \quad (7.22)$$

$$\sigma_{GMM}^2 = \boldsymbol{\rho}^T (\boldsymbol{\sigma}_S^2 + \boldsymbol{\mu}_S^2) - \mu_{GMM}^2 \quad (7.23)$$

$\boldsymbol{\mu}_S$ and $\boldsymbol{\sigma}_S^2$ are vectors with elements corresponding to predictions from each GP regressor with hyperparameter sample $\boldsymbol{\xi}_i$. Each of the mixing coefficients ρ_i are dependent on assumed knowledge $p(\boldsymbol{\xi})$ and, importantly, the form of the predictive surfaces of both $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi})$ and $p(\tilde{\mathbf{r}}|\tilde{Q}, \boldsymbol{\xi})$, as approximated by $v(\boldsymbol{\xi})$ and $\phi(\boldsymbol{\xi})$ (see [77] for the full definition of $\boldsymbol{\rho}$).

A strong feature of this model averaging approach is that the Marginal Likelihood $p(\tilde{\mathbf{r}}|\tilde{Q}, \boldsymbol{\xi})$ automatically assigns a high weighting to the GP regressors that explain the current active subset of measurements comparatively well, without the need for expert domain knowledge. As the active window moves across the workspace, the mixing coefficients of $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi}_i)$ freely adapt according to the changing workspace environment. In addition and unlike other approaches such as that of [85], there are no restrictions on the type of hyperparameters $\boldsymbol{\xi}$ that can be used in this framework so long as the covariance function is stationary (so that we may use the efficient implementation of Chapter 6).

7.3.4 Modifications to the Covariance Function

We now detail refinements to the covariance function $k(\mathbf{q}, \mathbf{q}')$ that we use in each of our stationary GPs $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi}_i)$. These modifications are so that we may exploit the

increased model flexibility afforded by the Bayesian Quadrature approach. Namely we decouple the length scale so that the underlying workspace surface may have differing degrees of smoothness in each dimension of sensor configuration space $\mathcal{Q}$, θ and t (see Section 4.4.1.2 for a discussion of smoothness). To create this anisotropic variant of the Matérn $k(\mathbf{q}, \mathbf{q}')$ we modify the Mahalanobis distance metric d of Equation 4.56

$$k(\mathbf{q}, \mathbf{q}')_{\text{matern } \nu=\frac{3}{2}} = \sigma_p^2 \left(1 + d\sqrt{3}\right) \exp\left(-d\sqrt{3}\right) \quad (7.24)$$

where

$$d(\mathbf{q}, \mathbf{q}') = \sqrt{(\mathbf{q} - \mathbf{q}')^T \begin{bmatrix} (l_\theta \psi_\theta)^2 & \mathbf{0} \\ \mathbf{0} & (l_t \psi_t)^2 \end{bmatrix}^{-1} (\mathbf{q} - \mathbf{q}')} \quad (7.25)$$

$\psi_\theta = 0.5^\circ$ and $\psi_t = \frac{1}{75}$ seconds are as per Section 5.2.⁵ This modification requires the substitution of two independent length scales $\{l_\theta, l_t\}$ in place of l (as commonly found in the GP literature [88]). Overall we therefore use three hyperparameters $\boldsymbol{\xi} = \{\sigma_p^2, l_t, l_\theta\}$ compared to the two $\boldsymbol{\xi} = \{\sigma_p^2, l\}$ in Chapter 6, one for range r , σ_p^2 , one for time t , l_t and one for laser beam angle θ , l_θ . We choose this augmentation to allow comparison with the results obtained using the stationary approach of Chapter 6 and as a simple demonstration of this non-stationary technique. Many other sensible choices for covariance function may be applicable to our problem domain. For example, we might boost the regression performance through the addition of a sinusoidal (or

⁵ ψ_θ and ψ_t are redundant scale factors that refer to the resolution of our laser scanner, kept only so that we may directly compare $\{l_\theta, l_t\}$ with l from Chapter 6.

periodic) covariance term (with amplitude λ , period ω and attenuation τ included in $\boldsymbol{\xi}$)

$$k(t, t')_{\text{periodic}} = \lambda^2 \exp\left(-\frac{\sin^2(d_t)}{2\tau}\right) \quad (7.26)$$

where

$$d_t(t, t') = \sqrt{(t - t')w^{-2}(t - t')} \quad (7.27)$$

The intent is to capture the cyclical artifact present in some sections of $\mathcal{Q}$ in t that is due to the rocking motion of the balancing platform used for data capture (see Chapter 2 and Section 3.4). τ provides a similar role as the length scales $\{l_\theta, l_t\}$ in Equation 7.25 that encodes the “forgetting” of covariance over time, and hence a degree of smoothness in predictions.

7.4 Simple Approximations to Predictive Bayesian Quadrature

In the previous section we discussed the theoretical building blocks that are required to apply Bayesian Quadrature to our regression framework. However, there are a number of practical considerations that are required before it can be applied to our problem domain. These stem from the covariance matrices of $v(\boldsymbol{\xi})$ and $\phi(\boldsymbol{\xi})$ (that approximate the posterior predictive distribution $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi})$ and the marginal Likelihood and $p(\tilde{\mathbf{r}}|\tilde{\mathcal{Q}}, \boldsymbol{\xi})$ respectively) being, like any GP, prone to poor conditioning. The effects of this poor conditioning are exacerbated by the noiseless observations of the distri-

butions that we are attempting to model (see Section 4.4.2) and the multiplication of these matrices (and the covariance of the prior $p(\boldsymbol{\xi})$) in the formation of predictions. As discussed in Section 4.4.2, poor conditioning is dependent on the assumed roughness of the underlying functions that we attempt to approximate and by the density of measurements in the input space of the regression. The additional *quadrature parameters* (which define $p(\boldsymbol{\xi})$, $v(\boldsymbol{\xi})$, and $\phi(\boldsymbol{\xi})$) and the positioning of our hyperparameter samples $\boldsymbol{\xi}_i$ therefore require special attention which we shall discuss further in Section 7.7. We aim here to make the necessary considerations that will inform such approaches whilst producing viable and principled approximations that incur as little computational overhead as possible. The resulting simple algorithms may then be used as a baseline to compare against future work to analyse the trade-off between increased algorithm accuracy versus computational requirements. As noted in Section 5.3 these approximations could also be used to self-select the family of covariance functions (and not just the hyperparameter values) that are most appropriate to predictions at the time. These approximations avoid the curse of dimensionality which would otherwise be difficult to circumvent with the full Bayesian Quadrature approach (see Section 8.1.2).

Concretely, we approximate the output of the GMM as a normal distribution (as per Equation 7.21) which we use as input for our active sampling strategy (Section 6.4). We use one of two alternative approximations for the mixing coefficients $\boldsymbol{\rho}$ that we detail in Sections 7.4.2 and 7.4.3 and visualise in Figure 7.1. The resulting algorithms require no additional quadrature parameters, they yield computational costs that are linear in the number of $\boldsymbol{\xi}_i$ and they are trivial to parallelise as an independent GP regressor can be maintained for each $\boldsymbol{\xi}_i$.

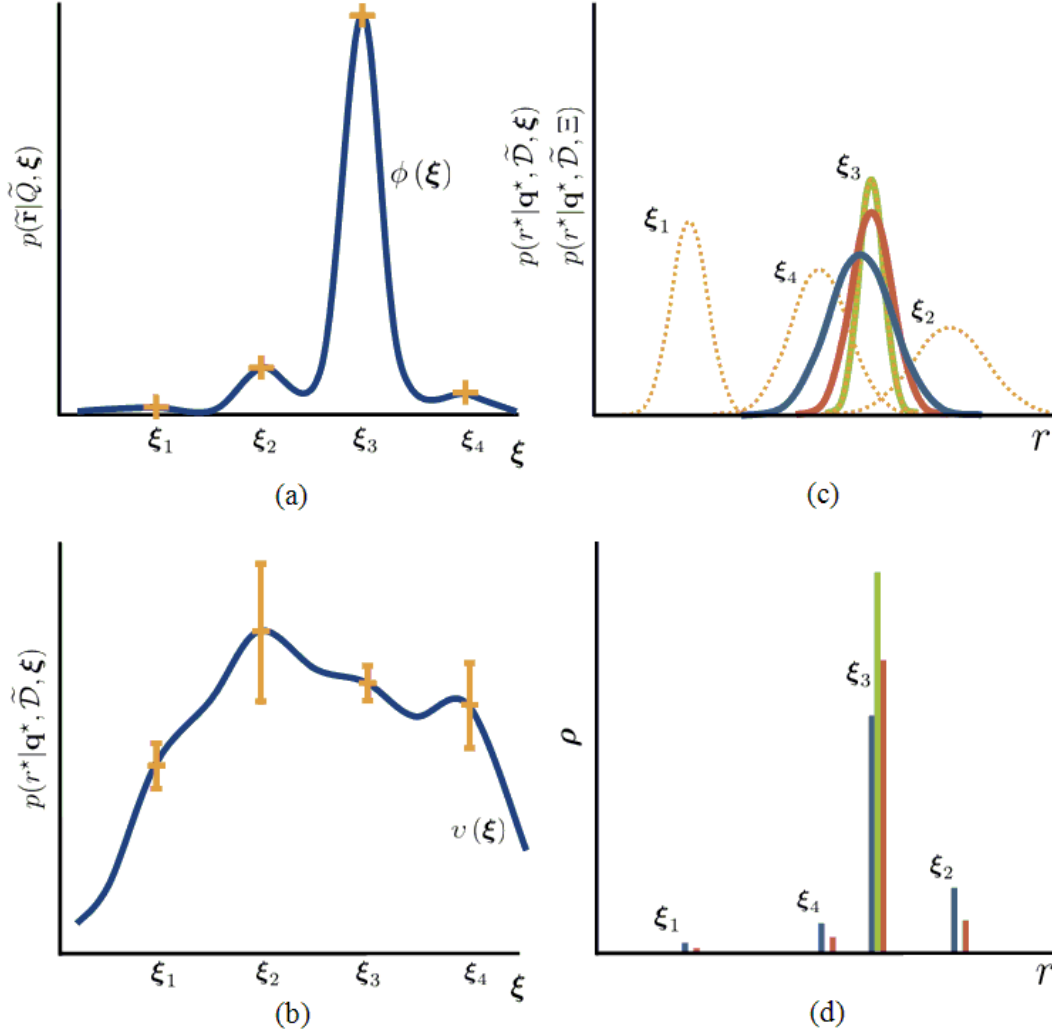


Figure 7.1: Visualisation of a prediction r^* using predictive Bayesian Quadrature and the approximations that are used in this thesis. For simplicity we have assumed a 1D hyperparameter sample space $\mathcal{E}$. Blue lines concern the predictions as made by the predictive Bayesian Quadrature of Section 7.3.3, red the Monte Carlo approximation of Section 7.4.2 and green the type-II Maximum Likelihood of Section 7.4.3. Orange indicates posterior predictive distributions $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \xi_i)$ from each of our stationary GPs predictions in plot (b) and (c) (with error bars that indicate the extend of their distributions), and their associated Marginal Likelihood of the measurements given hyperparameter sample ξ_i , $p(\tilde{r}|\tilde{Q}, \xi_i)$ in plot (a). Plots (a) and (b) concern hyperparameter sample space with the predictive surfaces $v(\xi)$, and $\phi(\xi)$ that correspond to the predictive Bayesian Quadrature approach. Plot (c) concerns predictions both from the individual stationary GPs $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \xi_i)$ and the final predictive distribution $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \Xi)$. Mixing coefficients ρ are depicted by the bar plot in (d). The type-II Maximum Likelihood uses only the stationary GP prediction with the greatest $p(\tilde{r}|\tilde{Q}, \xi_i)$, in this case ξ_3 , while the others use a combination of $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \xi_i)$. Since predictive Bayesian Quadrature attempts to capture the surfaces of $v(\xi)$, and $\phi(\xi)$ the overall effect is a lower weight on ξ_3 compared to that of the Monte Carlo approximation. As we shall discuss in Section 7.6 we have found that the majority of our predictions involve one highly weighted hyperparameter sample due to the roughness of the Marginal Likelihood surface as reflected in plot (a). In this instance all three approaches tend to produce similar results as shown in plot (c).

7.4.1 Choosing Hyperparameter Samples

In our GMM we weight multiple stationary GPs $p(r^*|\mathbf{q}^*, \widetilde{\mathcal{D}}, \boldsymbol{\xi}_i)$ in response to local workspace conditions. To permit the increased model flexibility that this non-stationary extension can afford over that of a single GP instantiation (Chapter 6) we must ensure that the set of GPs are collectively capable of capturing the diverse properties of the underlying surfaces. We do so by instantiating each GP with hyperparameter samples $\boldsymbol{\xi}_i$ that span a hypervolume of $\mathcal{E}$ (where $\boldsymbol{\xi}_i \in \mathcal{E}$) that is pertinent to the workspace (as found by our hyperparameter value investigation in Section 6.8.4). We assume no prior knowledge over the location of the samples within this volume and hence by the principle of indifference we use a bounded uniform distribution for $p(\boldsymbol{\xi})$. In this initial investigation we are interested in producing an algorithm with as little computational overhead as possible and therefore wish to use only a small number of hyperparameter samples $\boldsymbol{\xi}_i$. Rather than drawing samples from our proposal distribution $p(\boldsymbol{\xi})$ (as commonly found in Monte Carlo methods) we use an equi-spaced grid sampling of $\mathcal{E}$ to avoid biasing the coverage of $\mathcal{E}$.⁶

Additional expert knowledge about the prior distribution $p(\boldsymbol{\xi})$ may be incorporated through either modifying $\boldsymbol{\rho}$ (see Sections 7.4.2 and 7.4.3) or by manual, a priori, placement of hyperparameter samples $\boldsymbol{\xi}_i$. In the latter more $\boldsymbol{\xi}_i$ could be placed in regions of $\mathcal{E}$ where we require a greater weighting in the GMM and fewer where it is known that these areas are less pertinent to our predictions. This knowledge may be available, for example, when the sinusoidal term of Section 7.3.4 is included in $k(\mathbf{q}, \mathbf{q}')$ and typical

⁶This biasing would otherwise be exacerbated in the case of many hyperparameters due to the curse of dimensionality (where the space $\mathcal{E}$ grows exponentially in the number of hyperparameters).

values for vehicle periodicity are known. In the Bayesian Quadrature framework of [77] a variety of techniques are used to move individual ξ_i online according to the fit of the regressor to the current measurements which may also be incorporated here, albeit with added computational implications (see Section 7.7 for a further discussion).

For any GP regression there is a practical limit for the acceptable range of hyperparameter values to ensure that the resulting predictive surface does not overfit the training data. In our approach this restricts the choice of hyperparameter samples ξ_i .⁷ This is because each of our approximations of the mixing coefficients ρ (detailed in the following sections) are dependent only on a measure of the fit of our predictions to the training data (through the Marginal Likelihood $p(\tilde{\mathbf{r}}|\tilde{Q}, \xi)$). Figure 7.2 demonstrates the outcome when very low length scales l_t , l_θ and high process variance σ_p^2 are included in our set of hyperparameter samples Ξ in terms of surface reconstruction error.

⁷Note: an advantage of the the full BQ approach is that the prior $p(\xi)$ may be set independently to the hyperparameter sample locations. We may therefore use $p(\xi)$ to guard against overfitting and hence free our choice of hyperparameter sample placement.

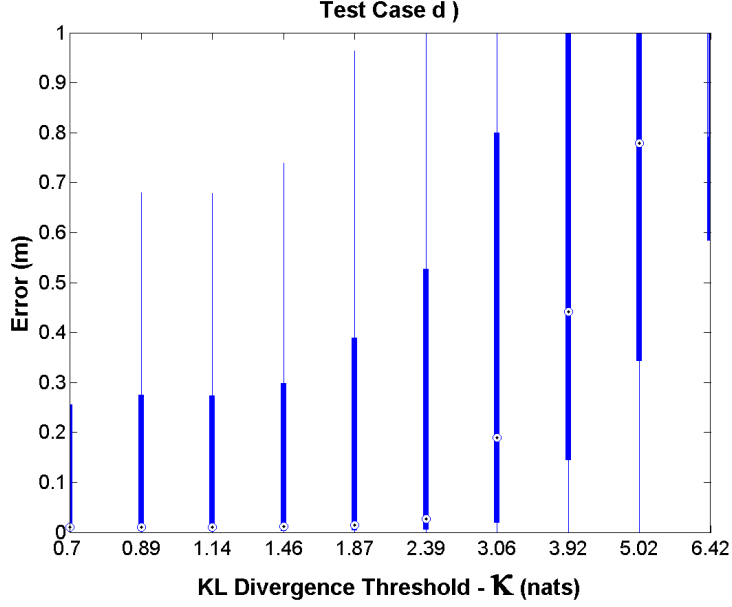


Figure 7.2: Demonstration of the sensitivity of this approach to the range of hyperparameters samples ξ_i . Box plots denote median and interquartile range of error for a selection of logarithmically spaced thresholds κ for test case (d) (see Figure 6.4). Whiskers are plotted that extend a maximum of one and a half times the interquartile range. By allowing our set of hyperparameter samples Ξ to include very small length scales $\{l_t, l_\theta\} = \{0.1, 0.1\}$ units and a very high process variance $\sigma_p^2 = 0.7 \text{ m}^2$ we produce measurement predictions that are highly overfitted. The resulting error statistics are poor compared to, for example, Figures 6.13 and 7.3.

7.4.2 Weighting Via Re-Normalised Marginal Likelihood

For regular grid spaced hyperparameter samples ξ_i we may use the following to affect a simple Monte Carlo approximation of Equation 7.2 using the GMM model given in Equation 7.21 (see Section 7.3.3)

$$\rho_i = \frac{p(\tilde{\mathbf{r}}|\tilde{Q}, \xi_i)p(\xi_i)}{\sum_{i=1}^g p(\tilde{\mathbf{r}}|\tilde{Q}, \xi_i)p(\xi_i)} = p(\xi_i|\tilde{\mathcal{D}}) \quad (7.28)$$

where both the marginal Likelihood of the measurements given hyperparameter sample $p(\tilde{\mathbf{r}}|\tilde{Q}, \boldsymbol{\xi}_i)$ and the prior over hyperparameters $p(\boldsymbol{\xi}_i)$ were introduced in Section 7.3. Through application of Bayes Rule (Equation 4.1) we may view each weighting coefficient as the posterior distribution of the hyperparameter sample given the data $p(\boldsymbol{\xi}_i|\tilde{\mathcal{D}})$. By assuming an uninformative bounded uniform prior $p(\boldsymbol{\xi})$ (see Section 7.4.1) we may write

$$\rho_i = p(\boldsymbol{\xi}_i|\tilde{\mathcal{D}}) = \frac{p(\tilde{\mathbf{r}}|\tilde{Q}, \boldsymbol{\xi}_i)}{\sum_{i=1}^g p(\tilde{\mathbf{r}}|\tilde{Q}, \boldsymbol{\xi}_i)} \quad (7.29)$$

7.4.3 Discretised Type II Maximum Likelihood (ML-II) Estimate

We also provide a simple model selection approximation that uses a single hyperparameter sample at any given time $\boldsymbol{\xi}_i$

$$p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \Xi) = p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \boldsymbol{\xi}_{ML-II}) \quad (7.30)$$

where:

$$ML-II = \underset{i}{\operatorname{argmax}} p(\tilde{\mathbf{r}}|\tilde{Q}, \boldsymbol{\xi}_i) \quad (7.31)$$

We have used the subscript symbol $ML-II$ to indicate that this is an approximate to the Maximum Marginal Likelihood using only a finite number of model hyperparameter

samples ξ_i .⁸

7.5 Non-Stationary Results

For ease of comparison with the stationary results of Chapter 6 we use the same experimental setup as detailed in Section 6.7 for each of our approximations. We use $\sigma_m^2 = 1 \times 10^{-4} \text{ m}^2$, and starting from the choice of stationary GP hyperparameter values $\xi = \{l = 8 \text{ units}, \sigma_p^2 = 0.05 \text{ m}^2\}$, we choose three, four and four equally spaced values in each of the $\sigma_p^2 = \{0.04 \dots 0.15\} \text{ m}^2$, $l_t = \{7 \dots 40\} \text{ units}$, and $l_\theta = \{7 \dots 40\} \text{ units}$ hyperparameters, respectively, creating $g = 48$ hyperparameter samples ξ_i in total.

7.5.1 Comparing Approximate GMMs

Figure 7.3 shows compression rate $\frac{|\mathcal{D}'|}{|\mathcal{D}|} \%$ across all test cases for our approximate GMM models.

Although the decrease in compression rate between the stationary (Figure 6.13) and the non-stationary (Figure 7.3) models are stark, we must also take into account the resulting error for a fair comparison. In Table 7.1 we collate an exemplar of roughly equivalent compression rate for each test case from Figures 6.13 and 7.3. We see that results are similar across all regression models. As we saw in the surface error plots in Figure 6.9 the main regions of error for the stationary model were around large discontinuities. Given that discontinuities occur relatively infrequently in our test cases it is reasonable that the change in algorithm performance is small. That said, the

⁸This Maximum *Marginal* Likelihood solution should not to be confused Maximum Likelihood solutions over model parameters ζ that apply to parametric models.

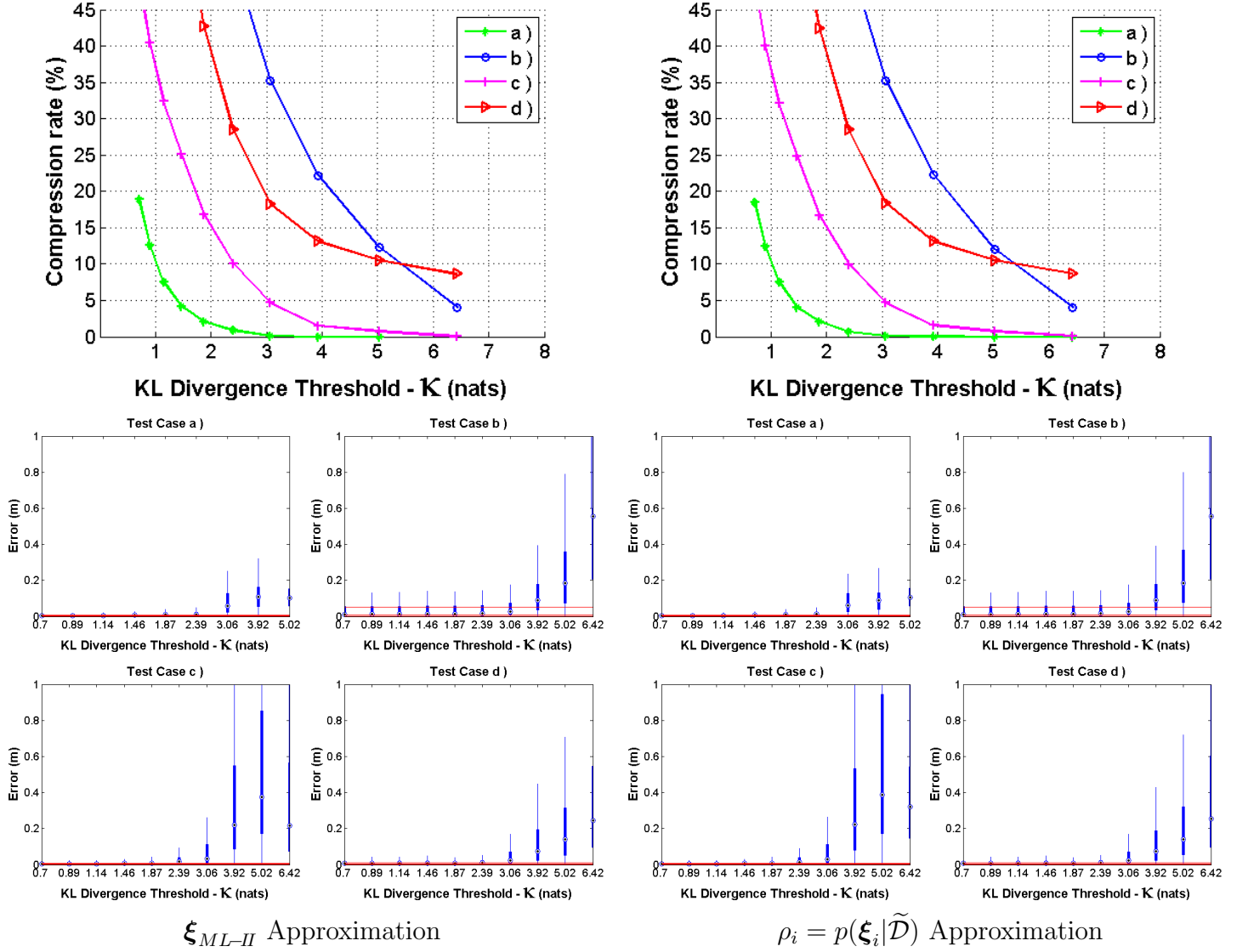


Figure 7.3: Summary plots of results using our non-stationary regression models for the test cases (a)-(d) (see Figures 3.10 and 3.11). Red lines indicate the median and interquartile range that result from using linear interpolation of all data points (as discussed in Section 3.5). They are provided to permit comparison to our approach. On the left are results for the maximum Marginal Likelihood ξ_{ML-II} approximation and, on the right, the GMM model $\rho_i = p(\xi_i|\tilde{D})$. (Top row) Compression rate $\frac{|\mathcal{D}'|}{|\mathcal{D}|}$ % versus prediction threshold κ nats. For a given κ , compression rate is far lower for both these approximations than in the stationary model (Figure 6.13). On comparing our approximations we see that they produce very similar results. Box plots (middle and bottom row) denote median and interquartile range for each of the κ indicated by markers in the compression rate plots (top row). Whiskers are plotted that extend a maximum of one and a half times the interquartile range. Again both approximations are very similar. To understand these results in more detail we therefore collate a small set of exemplars in Table 7.1.

increase in performance when using the non-stationary approach is significant, even when the stationary model uses more measurements (test case (b)). The difference between the non-stationary regression models are far smaller. For a given κ the re-normalised Marginal Likelihood approximation $\rho_i = p(\boldsymbol{\xi}_i|\tilde{\mathcal{D}})$ tends to use slightly more measurements, with only a slight increase in accuracy.

To understand the source of the increase in performance of the non-stationary models, we look at the active subset span η as the window moves across sensor configuration space $\mathcal{Q}$. In Figure 7.4 we plot η for both the stationary and the $\rho_i = p(\boldsymbol{\xi}_i|\tilde{\mathcal{D}})$ approximation (the maximum Marginal Likelihood $\boldsymbol{\xi}_{ML-II}$ approximation is very similar). We also depict the $\boldsymbol{\xi}_{ML-II}$ hyperparameter sample for each measurement ID, which indicates the changing workspace complexity. We see that the non-stationary η undulates more freely in response to changing workspace conditions than the stationary model. For test case (d) the non-stationary model is able to grow a larger η for the measurement IDs surrounding 1×10^4 (where the scene is less complex, as indicated by the longer length scales) than in the stationary case. In regions where the workspace complexity is changing dramatically (around measurement ID 2×10^4 in test case (a) and in the vicinity of 2.5×10^4 in test case (d)) we see that the non-stationary approach responds more fully by accepting more measurements into the active subset.

The final row of plots in Figure 7.4 indicate the number of significant GMM weights $\boldsymbol{\rho}$ for each prediction. We see that for test case (d) there is seldom a region where more than one hyperparameter sample $\boldsymbol{\xi}_i$ is relevant to predictions. This provides an insight to why both non-stationary models achieve similar levels of surface reconstruction accuracy, which we return to in Section 7.6. In the following section we examine the spatial

Test Case	κ (nats)	Stationary			
		$\frac{ \mathcal{D}' }{ \mathcal{D} } \%$	Median (m)	Mean (m)	Standard Deviation (m ²)
a)	2.399	6.88	0.00624	0.00818	0.00721
b)	3.927	36.69	0.0714	0.119	0.144
c)	3.069	4.17	0.0474	0.0946	0.126
d)	3.927	18.3	0.0406	0.0959	0.209

Test Case	κ (nats)	ξ_{ML-II} Approximation			
		$\frac{ \mathcal{D}' }{ \mathcal{D} } \%$	Median (m)	Mean (m)	Standard Deviation (m ²)
a)	1.145	7.53	0.00585	0.00734	0.00621
b)	3.069	35.3	0.0271	0.0653	0.112
c)	3.069	4.68	0.0321	0.117	0.188
d)	3.069	18.3	0.0235	0.0779	0.206

Test Case	κ (nats)	$\rho_i = p(\xi_i \tilde{\mathcal{D}})$ Approximation			
		$\frac{ \mathcal{D}' }{ \mathcal{D} } \%$	Median (m)	Mean (m)	Standard Deviation (m ²)
a)	1.145	7.58	0.00587	0.00728	0.00597
b)	3.069	35.3	0.0271	0.0653	0.112
c)	3.069	4.70	0.0309	0.108	0.169
d)	3.069	18.4	0.0239	0.0768	0.205

Table 7.1: Exemplars of results presented in Figures 6.13 and 7.3 of test cases (a)-(d) (see Figures 3.10 and 3.11). We select one exemplar for each test case and compare across all three regression models presented in this work. Exemplars are chosen so that compression rate $\frac{|\mathcal{D}'|}{|\mathcal{D}|} \%$ is roughly comparable between models. The three metrics are produced using the ℓ_1 norm in range between a hold out set of laser measurements and predictions that are made online as the actively sampled subset of laser measurements $\mathcal{D}'$ is established (as per Section 6.3.1). When compared with the stationary model, both non-stationary models provide a decrease in median error, even when the stationary model uses more measurements (test case (b)). Accounting for the slight increase in compression rate for the GMM $\rho_i = p(\xi_i|\tilde{\mathcal{D}})$ approximation, the two non-stationary models are very similar.

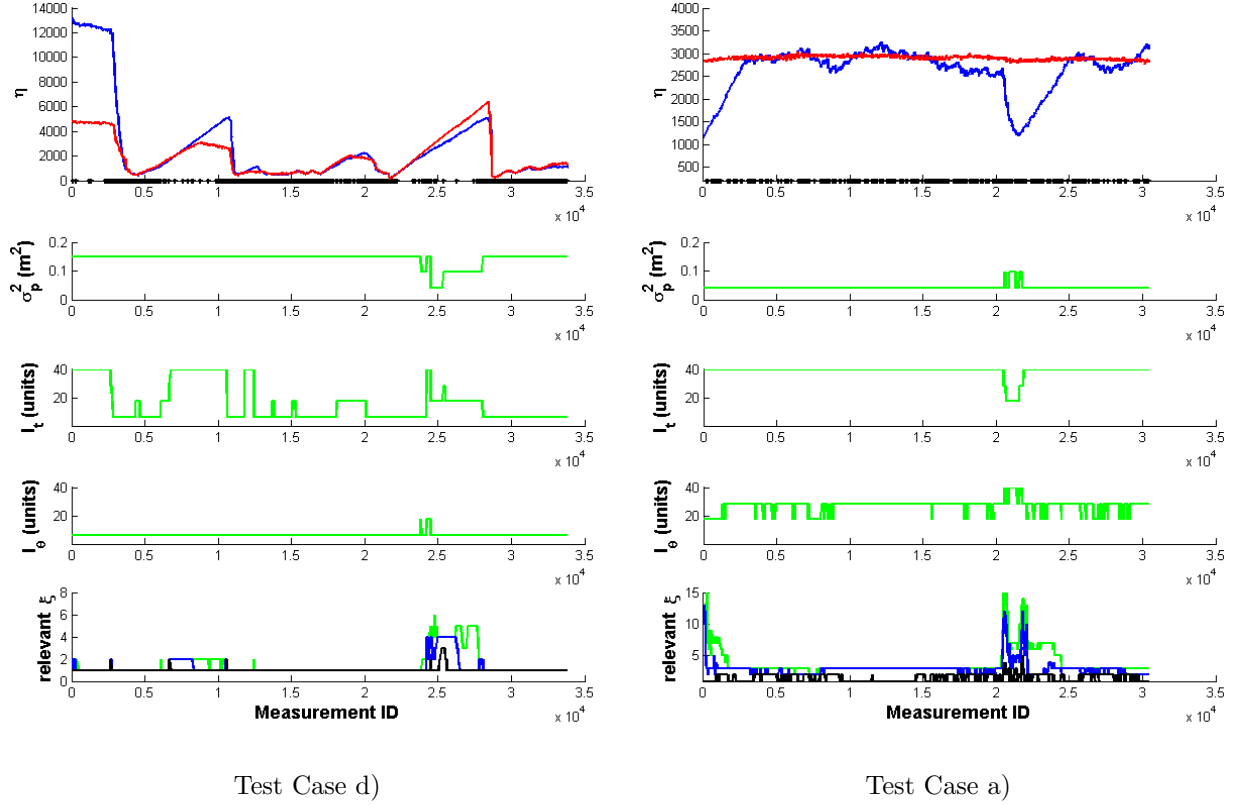


Figure 7.4: Visualisation of the active window as it moved across the sensor configuration space $\mathcal{Q}$. These plots correspond to the exemplars of test cases (a) and (d) in Table 7.1. Along the horizontal axes are the measurements as they are encountered by the active window as it progressed across $\mathcal{Q}$. We unwrap the two-dimensional $\mathcal{Q}$ surface in order to aid the comparison of measurement IDs across each of the plots below. Some of the jitter in all of the plots is associated with this unwrapping procedure. (Top row) Active window span η using the GMM $\rho_i = p(\xi_i | \tilde{\mathcal{D}})$ approximation in blue (with prediction threshold $\kappa = 3.069$ nats and $\kappa = 1.145$ nats from left to right). In red are the corresponding η for the stationary model ($\kappa = \{3.927, 2.399\}$ nats). Vertical lines along the horizontal axis indicate where measurements have been adopted into the active set $\tilde{\mathcal{D}}$ as a result of introspection for the GMM approximation (see Section 6.4). (Middle three rows) Hyperparameters $\{\sigma_p^2, l_t, l_\theta\}$ that are used in the maximum Marginal Likelihood ξ_{ML-II} approximation are provided to indicate how the workspace complexity changed across the test cases. (Bottom row) Plot of the number of hyperparameter samples with mixing coefficients $\rho_i = p(\xi_i | \tilde{\mathcal{D}})$ that are significant compared to ξ_{ML-II} . The green line indicates the number of hyperparameter samples that have a relative weighting greater than $\exp(-15) = 3.1 \times 10^{-7}$, blue $\exp(-10) = 4.5 \times 10^{-5}$, and black $\exp(-2) = 1.3 \times 10^{-1}$. First, we note that for the majority of each test case only a few hyperparameter samples ξ_i are relevant to predictions. This provides an indication as to why both non-stationary approximations achieve very similar results in Table 7.1. Second, by comparing the top four rows we see that the changes in η are highly correlated to the changes in the most highly weighted hyperparameter sample. We believe this is the reason for the relative performance of the non-stationary models compared with the stationary model.

distribution of the errors to illustrate the effects where multiple ξ_i become relevant.

7.5.2 Comparing Surface Error

We can view the distribution of surface error in 3D Euclidean space to understand the correlation of error with local workspace complexity as we did in Figure 6.9 of Section 6.8.2. Of particular interest, in terms of illustrative power, is the drain pipe, two thirds along the wall of test case (a) which we plot for each of our approximations in Figure 7.5. This corresponds to a dip in the active window span η in the vicinity of measurement ID 2×10^4 and a decrease in the time length scale l_t as shown in Figure 7.4. Overall, the error in Figure 7.5 progressively decreases between plots from the left to the right.

7.6 Discussion

We have discovered that the spacing between hyperparameter samples and the hypervolume over which the hyperparameter values varied was typically large compared to the smoothness of the Marginal Likelihood $p(\tilde{\mathbf{r}}|\tilde{Q}, \xi)$. This was observed by the small number (predominantly one) of hyperparameter samples ξ_i that were relevant to predictions across most of the workspace (see Figure 7.4). This resulted in both our approximations to the full Bayesian Quadrature approach producing very similar results in terms of compression rate and surface reconstruction error.

Given considerable additional computational cost, a greater density of hyperparameter samples would provide a greater selection of stationary GP models $p(r^*|\mathbf{q}^*, \tilde{\mathcal{D}}, \xi)$ and hence permit more accurate predictions for both models. Our Monte Carlo ap-

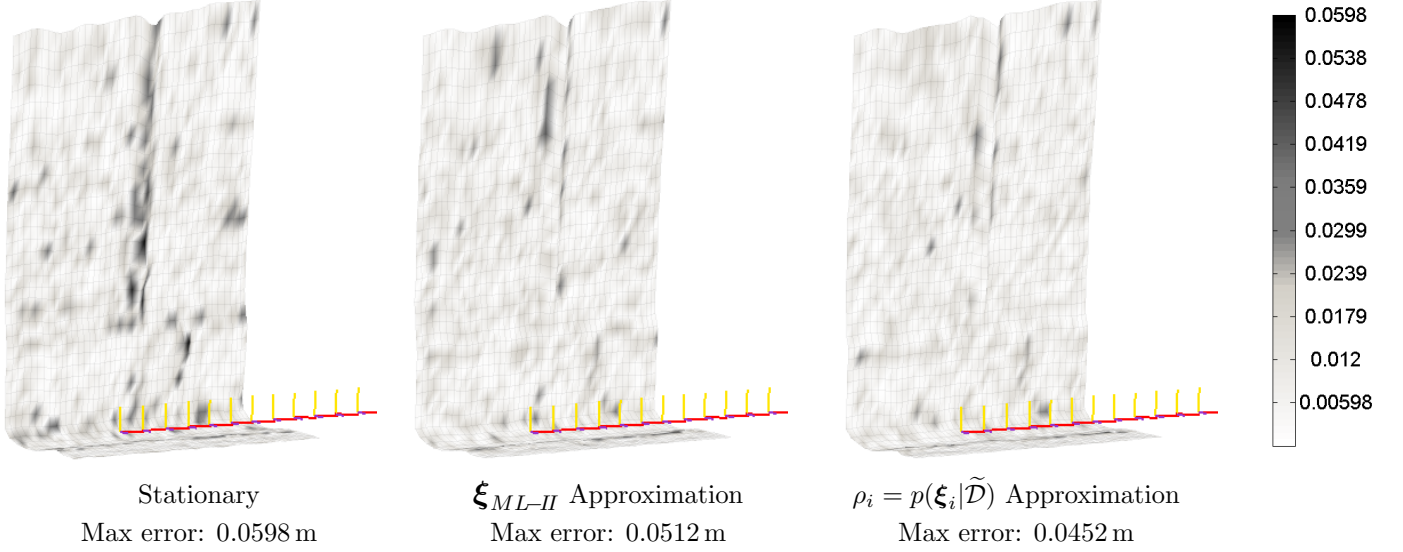


Figure 7.5: Error in surface predictions conditioned on the actively selected predictive support window $\tilde{\mathcal{D}}$ for each of our regression models. Predictions are made on-line as discussed in Section 6.3.1 and projected into Euclidean x, y, z space via the trajectory estimate of the sensor (see Section 2.4.4). Plots use prediction threshold $\kappa = \{2.399, 1.145, 1.145\}$ nats from left to right, corresponding to the exemplar for test case (a) in Table 7.1. For clarity only a short section of test case (a) that contains a drain pipe is shown in the figures. Colouring is provided by the ℓ_1 norm in range from hold out measurements and the predicted surface as discussed in Section 3.5. Colour scaling is the same for each plot to allow ease of comparison with black denoting high error and white, low. For the stationary model (left) we clearly see a change in colouring along the drain pipe. Colours are more evenly spread for our non-stationary models, indicating that these models have more accurately captured the relief around the pipe. When comparing non-stationary models one must take into account that the number of retained laser measurements $\mathcal{D}'$ is slightly larger when using the $\rho_i = p(\xi_i|\tilde{\mathcal{D}})$ approximation, as detailed in Table 7.1. The decrease in maximum error when using the $\rho_i = p(\xi_i|\tilde{\mathcal{D}})$ approximation are however significantly less, and the distribution is slightly more even.

proximation $\rho_i = p(\boldsymbol{\xi}_i | \tilde{\mathcal{D}})$ (Section 7.4.2) would particularly benefit from more informed predictions as more hyperparameter samples would be relevant at a given point in time. However, the full Bayesian Quadrature treatment provides an alternative approach that could lessen the computational burden. The full treatment uses the information contained in each hyperparameter sample more comprehensively (through using the location of each hyperparameter sample $\boldsymbol{\xi}_i$ in hyperparameter sample space $\mathcal{E}$) and could hence require less $\boldsymbol{\xi}_i$ for the same degree of prediction accuracy. This approach would also be capable of producing posterior predictions of hyperparameter values (see [77]) which may be useful for other processes such as vehicle trajectory estimation (see Section 8.1.5).⁹

In order for Bayesian Quadrature to be effective for our low sample density, strong smoothness assumptions are required for the GP approximations of the underlying quadrature functions $v(\boldsymbol{\xi})$ and $\phi(\boldsymbol{\xi})$ to be greater than zero across pertinent regions of our hyperparameter sample space.¹⁰ This is necessary because as the quadrature parameter length scales are decreased relative to the distance between hyperparameter samples, Bayesian Quadrature tends to our Monte Carlo approximation. However, as discussed above, a low *overall* sample density is unavoidable due to the large range in hyperparameter values (that are required by our workspace) and since only a limited number of hyperparameter samples may be used (to ensure an efficient implementation). Further, low quadrature parameter length scales for $\phi(\boldsymbol{\xi})$ are required in order to accurately capture the roughness of the Marginal Likelihood $p(\tilde{\mathbf{r}} | \tilde{\mathcal{Q}}, \boldsymbol{\xi})$ (as observed

⁹Our approximations already provide a coarse indication through the Marginal Likelihood $p(\tilde{\mathbf{r}} | \tilde{\mathcal{Q}}, \boldsymbol{\xi})$ of each hyperparameter sample $\boldsymbol{\xi}_i$.

¹⁰This phenomena has been observed in Section 6.5 where, in the absence of data, GP predictions of latent functions tend towards their assumed mean function (in this case zero $\boldsymbol{\mu}(\boldsymbol{\xi}) = \mathbf{0}$).

by only a few hyperparameter samples being relevant across swathes of the workspace in Section 7.5.1) and for the regression to be well conditioned (see Figure 7.1 for the effect that this has on $\phi(\boldsymbol{\xi})$).¹¹ To allow Bayesian Quadrature predictions to be significantly different to our Monte Carlo approximation we must therefore increase the *local* sampling density surrounding pertinent regions of hyperparameter space $\mathcal{E}$. This is so that there is sufficient support from multiple stationary GP predictions.

One technique that will increase local sampling density is to incorporate expert knowledge of the workspace through more a prescriptive placement of hyperparameter samples $\boldsymbol{\xi}_i$ in hyperparameter space $\mathcal{E}$. Expert knowledge could be acquired through an in depth analysis of the workspace by, for example, starting with an investigation similar to Section 6.8.4. This approach is applicable to both the approximations of Sections 7.4.2 and 7.4.3, and to Bayesian Quadrature. Alternatively a variety of optimisation techniques that reposition the hyperparameter samples $\boldsymbol{\xi}_i$ in accordance with the current measurements (by adding a third level to our hierarchical model) are suggested for the full Bayesian treatment in [77]. Each of these approaches require a careful practical consideration in order to avoid a substantial increase in computational requirements above that of the linear scaling with the number of hyperparameter samples $\boldsymbol{\xi}_i$ (see Section 7.3.2). For example, we could no-longer evaluate the static terms in the weighting coefficients $\boldsymbol{\rho}$ once, offline, as we could in see Section 7.3.2. Instantiating a new stationary GP (from moving or creating a hyperparameter sample $\boldsymbol{\xi}_i$) is also an $\mathcal{O}(n^3)$ operation (where n is the number of measurements in the active set) which would

¹¹As discussed in Section 4.4.2 poor numerical conditioning could be alleviated through approximations such as derivative observations and the addition of jitter (whereby an artificial measurement noise is included in the regression model $\sigma_m^2 > 0$).

increase the order of complexity of our overall algorithm.

Online assessment of the quadrature parameters ($p(\boldsymbol{\xi})$, $v(\boldsymbol{\xi})$, and $\phi(\boldsymbol{\xi})$) may also be required to ensure that conditioning is never problematic to our predictions. Here quadrature parameters could be re-optimised, constrained on producing well conditioned predictions, whenever predictions are poorly conditioned. Depending on computational capacity, additional quadrature hyperparameter optimisations may also be scheduled so that the shape of the marginalisation integrals may be accurately captured according to changing local workspace conditions. There are a number of techniques suggested in [77] that are capable of managing this task.

Our simple approximations and initial investigation provide us a baseline for us to compare against a full Bayesian Quadrature approach as future work. In particular, we will be able to analyse the trade-off in increased computational requirements versus algorithm accuracy.

7.7 Conclusions

This chapter provided a non-stationary extension to our treatment of the regression of laser data that was presented in Chapter 6. Our extension was motivated by the desire to allow different models to be used in accordance with their varying ability to model local regions of the workspace. We achieved this through weighting multiple stationary GP models of Chapter 6. This weighting was dependent on the Marginal Likelihood of the observed measurements in the current active window given each model. The active window could therefore adapt to workspace complexity more comprehensively which

resulted in a significant increase in performance of the non-stationary extensions over the stationary model.

To enable the non-stationary behaviour of our regression model, we extended the Bayesian treatment of the regression problem in Chapter 6 to marginalise over both model parameters and hyperparameters. Bayesian Quadrature ([73, 89, 77]) was used to approximate the intractable integrals and resulted in a hierarchical Gaussian Mixture Model. Two alternative algorithms were suggested that reduce the computation requirements of the full Bayesian Quadrature treatment through the exploitation of the Maximum Marginal Likelihood and our simple Monte Carlo approximations. These allowed the consideration of a large range of hyperparameter values with a manageable computational cost. Overall our algorithm scales quadratic in the size of the active set (as per Chapter 6) and linearly with the number of hyperparameter samples and is trivial to parallelise.

We used our initial investigation to inform the careful crafting of an algorithm that could be capable of implementing the full Bayesian Quadrature approach. In particular we noted the need to increase the local sampling density of hyperparameter samples above that used in our initial investigation. We then discussed further considerations that may also be required in order to produce a practical implementation which we leave as future work.

Chapter 8

Conclusions

In this thesis we have described a way to intelligently thin point cloud data that has been gathered from a push broom laser system. In doing so we provide access to the underlying workspace surfaces that are present in the laser data using a probabilistic framework. Specifically we have made the following contributions:

World’s first Robotics data paper (Chapter 2) - We designed, acquired and released a high-quality, peer-reviewed and free-to-access dataset to the robotics community. This release was predicated on a substantial amount of robotics engineering for the creation of a dynamically balanced data capture platform. The data paper has promoted the much needed sharing of robotics data and has stimulated further releases by the robotics community.

Sensor configuration space parameterisation (Chapter 3) - We adopted a novel parameterisation that could represent the non-functional relationship of workspace surfaces that exists between the ground plane and elevation of laser point clouds. This

parameterisation permits us to reform our surface estimation problem as one of regressing range measurements over sensor configuration space, which is amenable to Gaussian Processes. It also decouples vehicle trajectory estimates from that of our mapping algorithm thereby allowing the two to be computed independently.

Efficient implementation of Gaussian Process Regression (Chapter 6) - We ensure that our stationary GP framework is computationally bounded through forming predictions conditioned on a finite, time varying window of laser measurements in sensor configuration space. We also reuse computations whenever possible. The resulting algorithm has complexity that is quadratic (rather than cubic) in the number of measurements in the active window.

Active subsampling (Chapter 6) - By using the KL-divergence between laser measurements and their GP-prediction, and a technique that we call introspection, we arrived at a criterion for discarding redundant laser data. This criterion can be controlled via a single threshold parameter κ that sets the resulting trade off between surface error reconstruction and compression rate. Its value is entirely at the user's discretion.

Online and offline surface predictions (Chapter 6) - Our framework may be used online to actively sample the workspace as laser data is collected, whilst simultaneously providing predictions from the centre of the active window. Predictions may also be made offline using just the retained laser measurements. These surface predictions can be made *anywhere* on our underlying surface, rather than being restricted to the laser

data sample sites.

Parametric mean GP (Chapter 6) - We have followed [14, 74] in the use of explicit basis functions as the mean function of our stationary GP. By assuming infinite vagueness on the prior distribution of explicit basis functions we arrived at a formulation that required no additional parameters. This mean function is the key to achieving our compression rate as it permits our predictions to perform well even in the presence of sparse laser data.

Non-stationary GP (Chapter 7) - Although in Chapter 5 we suggested a GP covariance function that we have found pertinent to our laser acquisition system, our framework permits us to use *any* stationary covariance function. Our non-stationary extension is comprised of a bank of stationary GP regressors each weighted via two principled techniques that are inspired by Bayesian Quadrature. Each technique uses the Marginal Likelihood of the laser data given a model to adapt to local workspace conditions, and each produces a framework with complexity that scales linear in the size of the bank. We have also used our investigations to guide the implementation of the full Bayesian Quadrature approach as future work.

8.1 Future Work

In Section 7.6 we discussed the practical considerations for the production of the full Bayesian treatment of GP hyperparameters for push broom laser data. We made two suggestions that could allow the full approach to produce significantly different results to our approximations through an increase in the local hyperparameter sample density,

which was required for . These were to use either a more prescriptive initial hyperparameter placement or to add a third layer to our hierarchical model which would move hyperparameter samples around online. To alleviate numerical conditioning issues we suggested two approximations (derivative observations and the addition of jitter - see Section 4.4.2), and then discussed how the online assessment of quadrature parameters could also help to capture the unknown Bayesian Quadrature surfaces. The latter technique could guarantee that conditioning is never problematic. All these methods require careful consideration to ensure that the increase in computational burden is worth the improvements in accuracy above that of our current approximations.

There are also a number of practical considerations that could increase the efficiency of our framework which we discuss in Section 8.1.1. In Section 8.1.2 we then suggest extensions to our framework aimed at increasing the accuracy of our surface predictions. Considerations that are required in order to apply our framework to laser data generated from different laser acquisition systems are discussed in Section 8.1.3. We also consider how sensor configuration spaces may be joined together in Section 8.1.4, and possible uses for by-products of our framework in Section 8.1.5.

8.1.1 Implementation Speed-ups

Online active subset sizing - We are interested in the online assessment of the active set size n to reduce computational load where a small n will suffice. At present we fix n by arranging for the adoption and the disposal of measurements from our active subset to occur as an atomic unit. This is so that we can make use of in-place manipulations of our covariance matrix $K_{\tilde{\mathbf{r}}}$. Instead we could determine the support

size automatically. Here we would drop measurements that have a sufficiently small covariance with predictions that are made from the centre of the active window (as per our online prediction strategy of Section 6.6.4). Although this would require marshalling dynamically sized covariance matrices, this could provide a more efficient algorithm through ensuring that samples are kept in the active set only so long as they are useful to predictions.

Intelligent introspection policies - Whenever our algorithm adopts a measurement it launches an introspective pass upon all measurements currently not in the active set but within the span of the active set (see Section 6.4). A heuristic driven introspection policy could avoid making predictions for each of these measurements. For example, predictions could be made preferentially in the direction of greatest surface gradient, or even just every other measurement and then terminating on reaching the centre of the active window. This could dramatically decrease the computational requirements of these introspective passes (in the limit we could see a migration from that of the blue line to the lower red line in the timing plots of Section 6.8.3).

Infrequent polynomial basis function updates - The re-centring of our polynomial basis functions at every update to the active set (see Section 6.6.3) is unnecessary. They could instead be re-centred only when A becomes poorly conditioned. Checking the condition number of A (cubic in the number of basis functions – 10 in our implementation), could be less computationally expensive than recalculating all variables that are dependent on the polynomial basis functions. In our experiments in Chapter 7 we found that basis functions required re-centring once per example and only due to

the stationary GP models that had consistently low weightings ρ – that is approximately 1 re-centre per 1×10^6 updates of the active set. To ensure that this approach is compatible with our offline predictions (Section 6.3.2) we would however have to record whenever basis functions are re-centred, although this storage overhead would be small.

Explicit online surface visualisations - Our surface predictions are presently rendered by interpolating the mean of all the online predictions of our implicit surfaces in a batch process. Rendering these online as the active window progresses through the workspace would not be a difficult task. We could reduce the triangle count of our surface meshes through making the triangle size a function of the span of the active window. Here larger triangles would represent smoother sections of the workspace surface and smaller triangles will arise around areas of the workspace where the active window span has collapsed due to complex surface geometries (see Figure 7.4).

8.1.2 Alternative Models

Range dependent measurement models - Throughout our investigations we have assumed a fixed measurement noise in our regression model. Although we found in Section 6.8.5 that globally varying this value has little impact on our regression, we saw that an increase in accuracy and a decrease in compression rate if we allowed measurement noise to increase with range. This would permit our framework to more accurately model the real physical process that is occurring during data collection. Measurement noise appears along the leading edge of our covariance matrix $K_{\mathbf{F}}$ and in the measurement model that is used in our active sampling strategy. It would be trivial

to implement these modifications once the new range dependent model is created.

Heteroscedastic measurement models - We may also wish to allow our measurement model to vary according to the location of the measurement in sensor configuration space $\mathcal{Q}$. For example, we know that because of the geometry of our data acquisition system, laser measurement with low values of beam angle θ are likely to measure the distance to the floor (see Section 3.4). One could construct a simple laser model of the varying oblique angles between the laser beam and the surface normal of the floor. We know that laser scanners generally produce noisier measurements the greater this angle and this is explored in detail in [84].

Alternative basis functions - We chose a simple polynomial regression model for the mean function of our stationary GP (Section 6.5), but others may be more appropriate. Alternative formulations could also provide computation savings by allowing the update of variables that are dependent on the basis functions, rather than having to recalculate them each time the active subset is updated.

Data-driven covariance function selection - In our non-stationary extension of Chapter 7 we permitted our GP framework to vary in response to local workspace conditions. In our investigations we used a range of hyperparameter values, however there is nothing stopping us from including a variety of covariance function families (Squared Exponential, Matérn, Exponential etc.). Here our simple approximations provide a computational advantage over the full Bayesian Quadrature approach as they naturally handle disjunct hyperparameter spaces (one for each covariance function family). The full Bayesian Quadrature approach would require a mapping to relate the hyperparam-

ter spaces (again GPs could be used for this). Alternatively a single covariance function could be used with components that encode each of the different covariance function families of interest. The latter risks an explosion in the number of hyperparameters; meaning it would be difficult to ensure sufficient local hyperparameter sample density due to the curse of dimensionality.

8.1.3 Alternative Laser Sensor Configuration Spaces

Tiled sensor configuration space - Our current formulation uses an active window that spans all sensor beam angles (the angle dimension θ of sensor configuration space $\mathcal{Q}$). Whilst this is feasible for our SICK based laser acquisition system, the size of the active window could become prohibitive for higher resolution systems such as those produced by for example FARO, Reigl and Leica. Independent active windows that collectively span θ may be sufficient. Jumps in predictions along boundaries could be avoided by weighting predictions from each active window according to the distance from the centre of each active patch (following the Gaussian Mixture Model approach of Section 7.4.2). We could alternatively overlap our active windows so that predictions near window boundaries have sufficient support. This technique has proved effective for the static windows of [85]. To reduce computational requirements we may be able to emulate this behaviour with non-overlapping windows that have just a few additional measurements that are judiciously chosen from neighbouring patches.

Adaptive windows - We may increase the accuracy of our predictions if we also permit each active windows to vary in shape in both dimensions of sensor configuration

space. Segmentation techniques such as those used in [99] could be used to drive the partitioning of laser data so that each active window has measurements with homogeneous surface properties. The KL-divergence (as used in our active sampling strategy - see Section 6.4) between each measurement and corresponding predictions from each active window could also be used to determine active window ownership. This latter technique would allow a degree of outlier detection when no active windows accurately predict a measurement. However, these techniques would require the careful management of the active windows especially on their creation and termination to ensure that there is sufficient support for predictions.

Increased dimensionality of sensor configuration space - As mentioned in Section 3.3 we may also parametrise alternative laser acquisition systems through trivially increasing the dimensionality of our sensor configuration space (for example for flash LIDAR systems and those manufactured by Velodyne - see Figure 8.1a). Multiple laser systems (see Figure 8.1b) could also be parametrised either via a single shared sensor configuration space, or linked spaces. The latter representation could use the work of [78] where measurements from one space inform predictions of the others since the same underlying workspace surfaces would be observed by all lasers, but at different times. This could help dramatically decrease compression rate of the overall system and possibly provide cues for dynamic object detection.



(a)



(b)

Figure 8.1: Alternative laser acquisition systems. (a) Velodyne HDL-64E. A head of 64 lasers rotate around an axis. Laser data could be parametrised by one time dimension and two angle dimensions (one for head rotation, and another for angle from rotational axis). (b) Multiple plane sweeping lasers mounted on a rotating disk. Here laser data could use the sensor configuration space of (a), shared by all three scanning units. Alternatively each unit could use a separate identical space.

8.1.4 Combining Sensor Configuration Spaces

One can appreciate that a tiny compression rate could be achieved if we re-used the set of retained measurements $\mathcal{D}'$ from one pass of a workspace to inform predictions on subsequent passes. By comparing measurements from each pass of the workspace (through application of the KL-divergence used in Section 6.4) we could also detect dynamic objects in the scene, or inconsistencies between regions of sensor configuration space. However, this requires there to be a mapping between relevant parts of sensor configuration space. This is in general a non-trivial task due to lack of perfect trajectory estimation. In the following we make three considerations that may assist this task.

Homogeneous Trajectories - Sometimes it is known in advance that the trajectory

of the laser scanner will be similar on each pass of the workspace. Passageways, for example, restrict motion to a narrow corridor. Likewise do road networks combined with ackerman steering for automobile mounted sensors. Under these scenarios it may be sufficient to simply use the velocity profiles and start and end times of the constrained segments of travel in order to map between the corresponding regions of sensor configuration space.

Heterogeneous Trajectories - Otherwise regions of sensor configuration space could be chosen that have underlying workspace predictions that are consistent in the global Euclidean frame. Our method used in Section 3.2.1 could be employed to detect these regions and inform the mapping.

Combining Sensor Configuration Spaces Through Pose Relaxation - Alternatively, laser data could be used to hone trajectory estimates with the aim of increasing the global consistency of point clouds. Here additional pose relaxation constraints could be added to the robotics system that was described in Section 2.4.4. These constraints could be derived from the alignment of corresponding laser measurements between distinct passes of the same workspace (possibly using descriptors such as spin images [45]). The result would be a non-rigid transformation of the laser point cloud (as parametrised by the vehicle trajectory) that could reduce the bending and misalignment issues highlighted in our discussion of Section 3.2.2.

8.1.5 Useful Framework By-Products

Iterative Closest Point (ICP) improvements - It is noted in [92] that ICP can produce more accurate and robust alignments when the point cloud has been pre-filtered to ensure that there is an even distribution of surface normals. As discussed in Section 6.4 our active sampling strategy chooses only measurements that are salient to the production of accurate surface predictions. These measurements typically lie on complex non-planar surfaces or at intersections between surfaces thereby ensuring a more even distribution of surface normals than in the original data. We have observed this in, for example, Figures 6.3, 6.4 and 6.5 and by the variability in the span of the active window extent η (and therefore the retained sample density) in response to local workspace complexity in Figure 7.4.

Hyperparameter posterior predictive distribution - Other processes could use the hyperparameter sample weightings that are as a result of our approach, to refine their inference in, for example, vehicle trajectory estimation. This ability could be particularly beneficial when covariance functions are used with hyperparameters that represent physical aspects of the vehicle motion (such as the periodic covariance function we suggested in Section 7.3.4). A full Bayesian Quadrature implementation would also permit a posteriori prediction of the most likely hyperparameters (as mentioned in Section 7.6), rather than just a weighting of the discrete values that are produced by our current algorithm.

Rapid stationary GP assessment - The non-stationary extension of Chapter 7 could be used to perform a rapid assay of appropriate hyperparameter values for a

specific sensor configuration space. Here statistics on the stationary GP weightings ρ would be collected over time. These could indicate which of the stationary GP models would be the most appropriate for the task at hand using only a single pass of the data (thereby avoiding the grid search that we performed in Section 6.8.4). Only the single stationary GP with consistently highly weighted ρ would then be used when computational capacity is limited.

8.2 Concluding Remark

In this thesis we have presented a novel approach to the 3D laser mapping and compression problem. Our approach provides access to underlying workspace surfaces whilst decimating point clouds in excess of two orders of magnitude without significant degradation in fidelity. Moreover we have examined all the necessary theoretical aspects that are required to implement our windowed GP framework for a push broom laser system. There are a number of promising extensions that can both further increase the efficacy of our approach, and see its application to alternative sensor acquisition systems. We hope that this work can facilitate the representation of increasingly large and complex outdoor scenes and stimulate further research into mapping with general surfaces.

Appendix A

Research Platform Details

This appendix describes additional information about our research platform which we introduced in Section 2.1.

The platform is comprised of two independently powered subsystems, a drive unit (a Segway RMP 200 research base) and a sensor payload. The drive unit uses its wheel encoders and internal IMU (which provides full roll, pitch and yaw information) to ensure that the research vehicle remains dynamically balanced whilst in motion. It is controlled via a bi-directional CAN-bus link which encodes desired rotational velocity for each of the two wheels. Within the dural mainframe of the sensor payload there are:

- two lithium ion battery packs (capable providing power for 90 minutes)
- four small form factor 2.4 GHz 100 Mbits/s networked PCs with 1 TB storage
- a remote operation module

- all interface and power-conditioning hardware for the platform’s sensor suite.

Each sensor is highlighted by a colour box in Figure A.1 and was described in Table 2.1. The platform can be controlled either via the remote operation module (which uses standard hobby equipment for remote-controlled aeroplanes) or via a laptop that is connected to the platform’s wireless ethernet bridge. Control signals are parsed by the platform’s primary PC and communicated to the drive unit via the CAN-bus.

There are a number of safety critical systems that are used to ensure safe operation of the research platform. This is important for humans as well as the on-board sensors – the vehicle can travel at 12.5 mph and weighs over 100 kg. These systems are implemented both in software, as user checks and warnings, and in hardware to provide fail-safe means to bring the robot to a halt. The four safety bumpers, that can be seen protruding from the front and back of the vehicle (Figure A.1), absorb the impact of any sudden and unexpected halting of the robot. There are also four safety cut-out buttons directly wired to the drive and payload sub-systems, and a CAN-bus breaker that acts as a *dead-man’s switch*, should the robot go out of range of its operators. If this happens the link between the drive unit and the sensor payload is severed which triggers the vehicle to be brought to a safe, balancing, halt. We also constructed a breakout of the drive unit’s user interface, as we shall discuss in the following section.

A.0.1 Finishing the Robotics Research Platform

At the start of this research the robotics platform was under construction. Although the main dural framework and most of the electrical power-conditioning and PCs were in place, there was a substantial amount of systems engineering involved in order for the

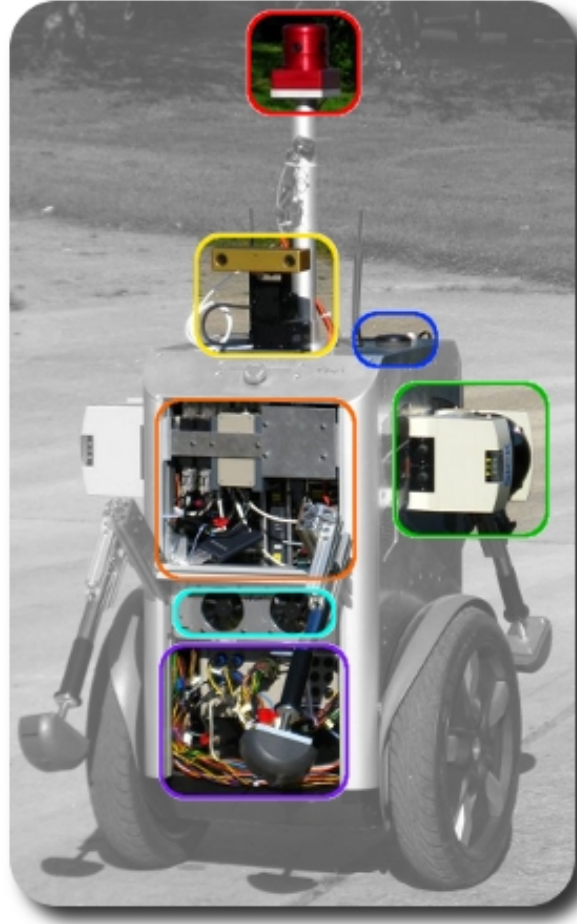


Figure A.1: Robotics research vehicle with sub-components highlighted in coloured boxes. Yellow, red, green and dark blue boxes are the stereoscopic camera, panoramic camera, left laser scanner and GPS receiver, respectively. For more information concerning these sensors see Table 2.1. The circles in the light blue box are the air inlet for the fan cooling system which is required to stop the small form factor PCs from overheating. Visible in the orange box are the housings for: the ethernet switch, the Laser serial to USB converters and the CAN-bus breaker. Behind these are the two battery packs and the four small form factor PCs. Within the wheel arches of the vehicle (the purple box) are the drive unit's control circuitry and the sensor payload's power box. Inside the power box are four buck-boost converters that supply various DC voltages to the sensors and the PCs via a relay board that is controlled by the primary PC. Careful considerations were made to ensure that the vehicle could balance under the heavy weight of the sensor payload – the vehicle weighs over 100 kg in total.

platform to begin collecting data. The author took the lead role in in this task, which occupied the first 18 months of this DPhil. Of particular note is the safety breakout of the drive unit’s user interface. This breakout was essential as it provided a means to turn the drive unit on, and to disarm its balancing circuitry without reaching into the bowels of the vehicle. This required the piggybacking of the existing user interface, and the electrical shielding and inverting of all signals before sending them to the top of the vehicle to minimise the risk of damage to the drive unit’s circuitry. We powered the board from the drive unit’s control batteries to avoid maintaining an unnecessary third power system. See Figure A.2 for details.

A.0.2 Software Considerations

We chose to use the Mission Orientated Operating System (MOOS) [68], depicted in Figure A.3. This system allows each software module to operate independently whilst ensuring that all data streams are synchronised to three decimal places of accuracy (by using a predecessor of the TicSync software [37]). As modules were authored by different group members, module independence was important in facilitating a simple and extensible development environment that could operate even in the face of multiple module failures. The MOOS specifies a client-server paradigm where all communications between components travel through a central server (called *MOOS DB*) over the sockets protocol in plain text. This paradigm has been broken for components on *Lisa 4* (Figure A.3) where high volume video output from the stereoscopic head requires a more direct connection to its consumer.

Verbose and human-readable logging proved invaluable in the deciphering and isola-

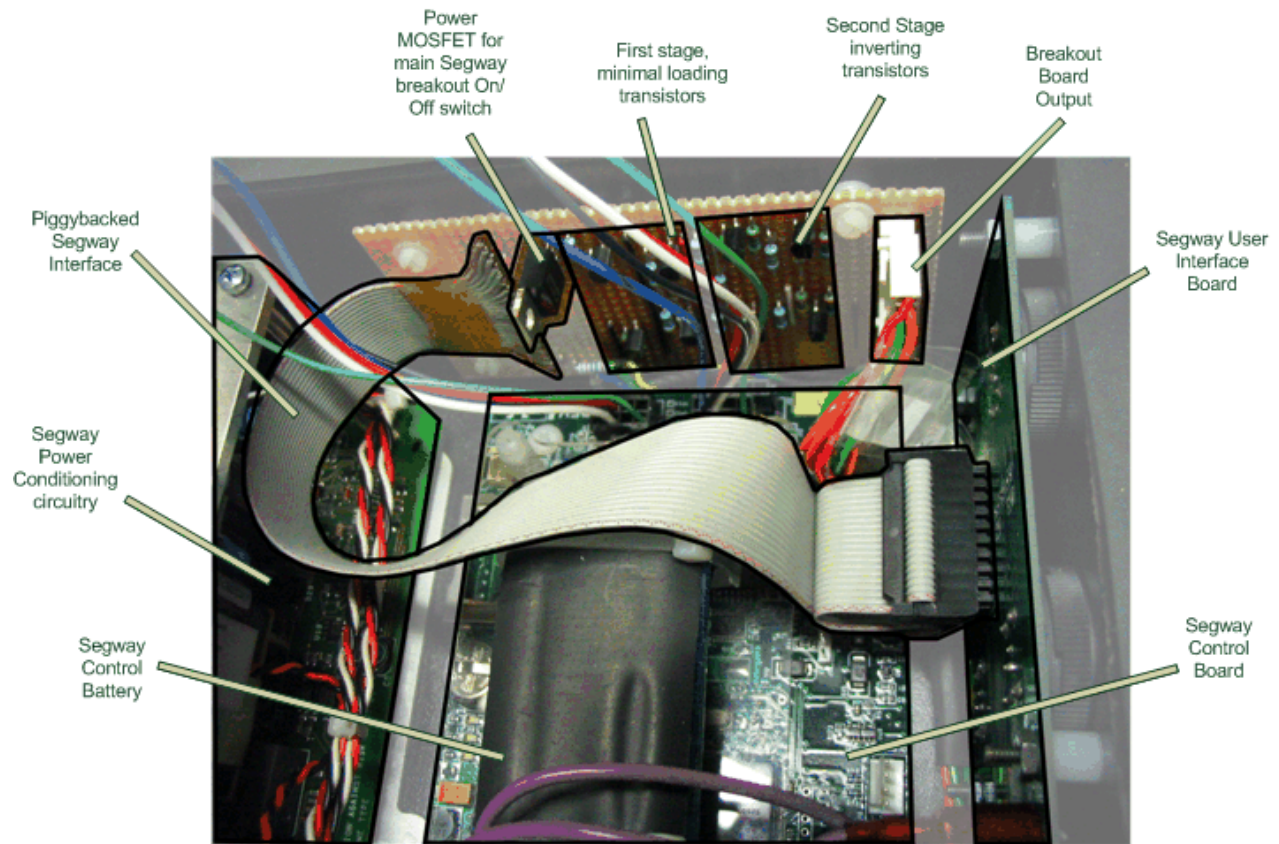


Figure A.2: Annotated photograph of the safety breakout board installed within the drive unit's control box. The control box is buried deep behind the wiring in the purple box of Figure A.1, which meant that this breakout board was essential for ensuring that the vehicle could be operated safely.

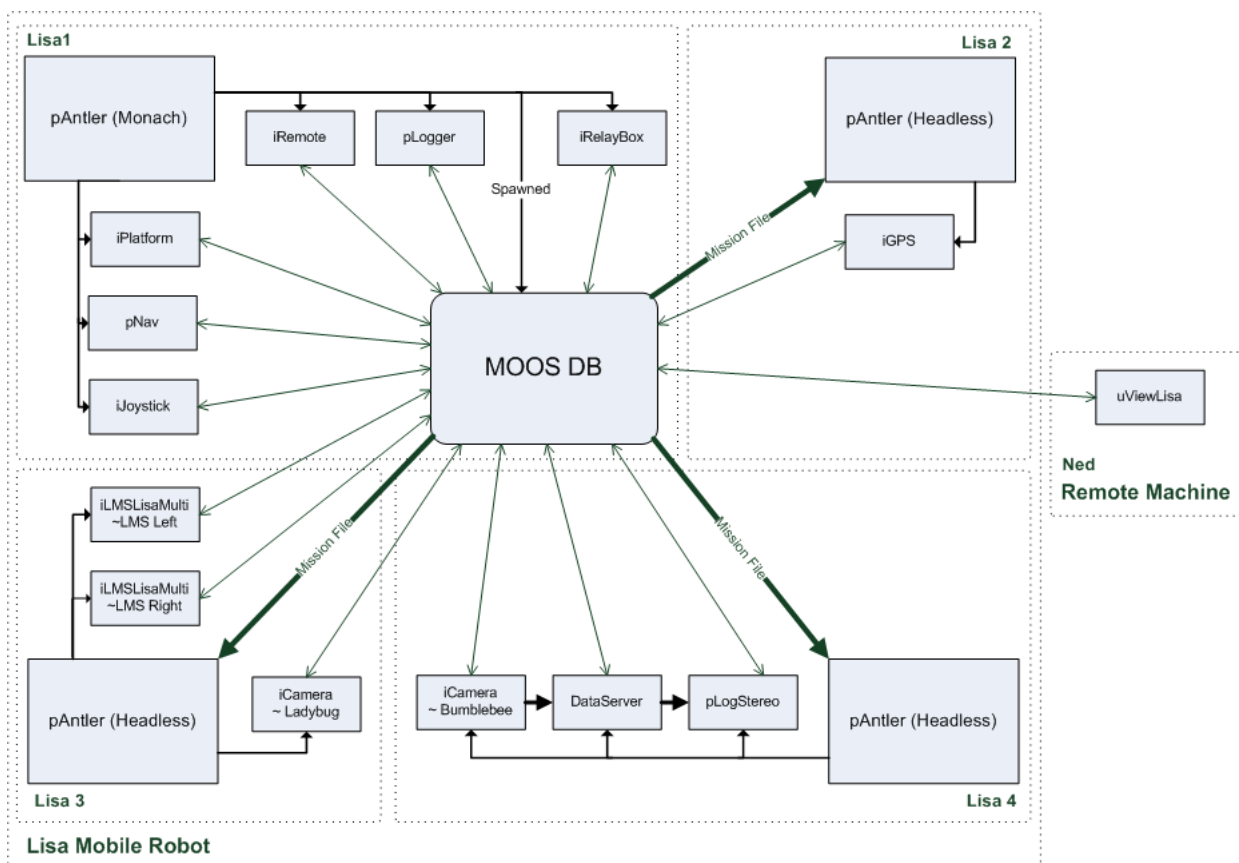


Figure A.3: Typical software topology for our research platform, as used throughout this dissertation. Each blue filled box represents a software module. Chained boxes that are labelled *Lisa 1* through *Lisa 4* indicate which of the four PCs (mounted inside the research platform) that each module runs on. *Ned*, a wirelessly connected laptop, was used to monitor and control the research platform. Each PC continually runs *pAntler*, which is responsible for initialising and reporting on the status of each module to the central server *MOOS DB*. System start-up is initiated by *pAntler* on *Lisa 1* which spawns the *MOOS DB* as well as other local software modules. *MOOS DB* then publishes a thinned version of the mission script (which it has received from *Lisa 1*'s *pAntler* on start-up) to the *pAntlers* running on the other PCs. These PCs then spawn their own local software modules. Only system-critical modules run on *Lisa 1* to ensure timely handling of important data. Once spawned each module communicates directly with the *MOOS DB*. Modules that have high data rate requirements additionally have direct interfaces between software modules (indicated by the thick arrows between *iCamera ~ Bumblebee*, *DataServer* and *pLogStereo*). *uViewLisa* uses the drive unit's odometry and laser data to render point clouds remotely on the fly as data is captured.

ting of software bugs. For example, by logging the drive unit's systems we determined that voltage dips in the control battery (due to the battery becoming exhausted) led to silent and immediate system shutdowns irrespective of whether the robot was travelling at speed. For a balancing robot this presented not only erratic behaviour, but also meant that the power circuitry for the motors overheated and failed, which required expensive replacement in New Hampshire, USA. Through reading the *MOOS DB* log we determined the source of this bug and wrote preventative code that monitored the drive unit's control battery voltage levels. Warnings were issued by the MOOS user interface to the vehicle operator whenever the battery voltage fell below a safety threshold. If the voltage fell below a further threshold the vehicle was brought to a safe halt.

Appendix B

Pose Relaxation

Motion estimates from different sensor modalities often disagree. Ideally disagreements are due to sensor noise, however they frequently occur due to inaccuracies and are sometimes caused by sensor failure. Pose relaxation similar to [69] is used in our framework to reconcile gross inconsistencies between sensors. Inconsistencies are particularly apparent surrounding loop closure events where combined errors in Visual Odometry, laser scan matching and visual registration can accumulate along a long kinematic chain of robot poses. Figure B.1 illustrates the typical structure of the pose graphs generated by our SLAM system (as introduced in Section 2.4.4).

Several authors have examined methods for pose graph relaxation in recent years [33, 105]. The goal is to perturb, or “relax”, the edges of the graph to accommodate, in a minimum error sense, all relative transformations to produce a coherent metric map. To construct the problem in a manner amenable to classical non-linear optimisation techniques we define $V = [v_1^T, v_2^T \dots]^T$ as the state vector we wish to optimise. v_i is the

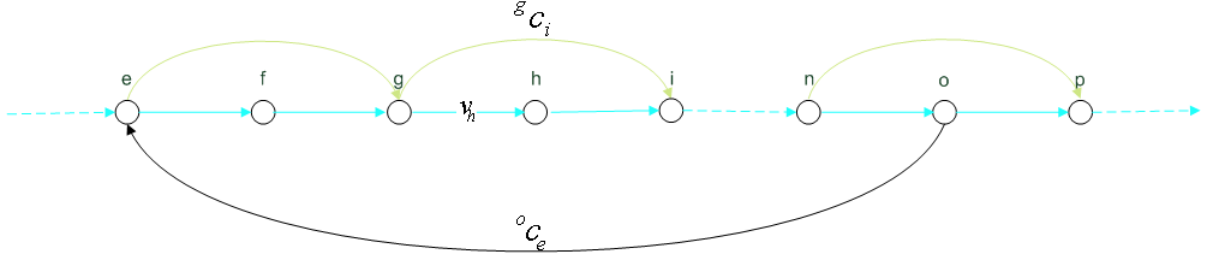


Figure B.1: A section of typical pose graph. Poses $(e, f \dots p)$ are denoted as nodes (circles) and edges are relative transformations. There is a chain of relative transformations flowing through the graph created by Visual Odometry (blue). Where possible laser scan matches (green) and visual registration (black) from loop closing events add additional constraints. Additional constraints ${}^i c_j$ are single edges linking disparate nodes (i and j) of this chain.

relative transform from pose $i - 1$ to pose i . Consider Figure B.2 which shows a general constraint between two poses m and q . The transformation, ${}^m T_q$ between two poses m and q is the composition ($\oplus$) of all the individual transformations between poses

$${}^m T_q = v_{m+1} \oplus v_{m+2} \cdots \oplus v_q \quad (\text{B.1})$$

${}^m T_q$ constitutes a prediction of the constraint ${}^m c_q$ supplied by our pose graph and $d_M({}^m c_q, {}^m T_q)$ is a measure of the compatibility between it and the constraint, with subscript M to indicate an uncertainty weighting on each of the elements of the pose. We shall split the constraints into n laser scan matching and loop closure events, and m interpose Visual Odometry measurements (for Visual Odometry ${}^m T_q$ becomes simply v_q as shown in Figure B.1). If we denote the stacked vector of Visual Odometry estimates as $O = [o_1^T, o_2^T \cdots o_m^T]^T$ and all scan matches and loop closures as $C = [{}^{a(1)} c_{b(1)}^T, {}^{a(2)} c_{b(2)}^T \cdots {}^{a(n)} c_{b(n)}^T]^T$ (where a and b are look up functions to produce the correct start and end poses of the kinematic chain), the error metric we wish to impose

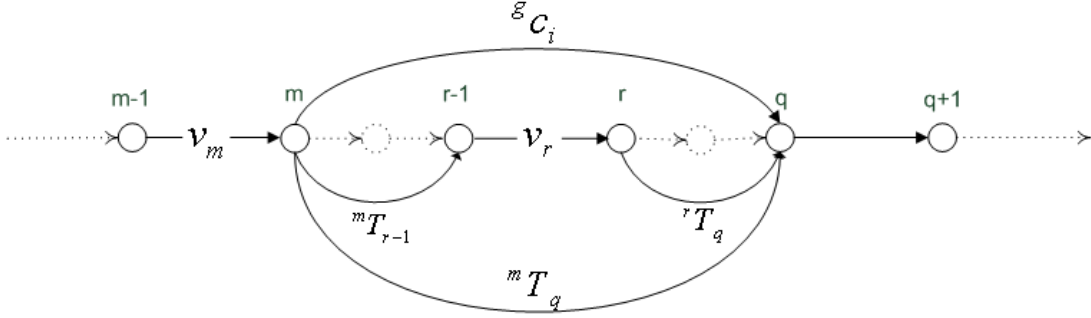


Figure B.2: A general section of pose graph showing a loop closure between vehicle poses m and q and a state of interest v_r . Note the pose graph optimisation is over transformations between vehicle poses and not the poses themselves. The dotted circles represent an arbitrary number of poses.

on the whole graph is

$$e(V|C, O) = \sum_i^n d_M(a^{(i)}_{c_{b(i)}}, a^{(i)}_{T_{b(i)}}) + \sum_j^m d_M(o_j, v_j) \quad (\text{B.2})$$

To minimise this error function we typically require the equation derivatives with respect to the state vector V being optimised. Remembering Equation B.1 and referring to Figure B.2 we must therefore consider incremental changes in each mT_q (where we have used the substitution $r = m + 1$).

$$\delta {}^mT_q = \sum_{r=m+1}^q \frac{\partial {}^mT_q}{\partial v_r} \delta v_r \quad (\text{B.3})$$

where δv_r is an incremental change in the r_{th} component of the state vector V (i.e. the relative transform between pose $r - 1$ and pose r). Using B.1 the partial derivative in the summation can be written in the following form

$$\frac{\partial {}^m T_q}{\partial v_r} = \frac{\partial \{v_{m+1} \oplus v_{m+2} \cdots \oplus v_q\}}{\partial v_r} \quad (\text{B.4})$$

$$= \frac{\partial \{{}^m T_{r-1} \oplus v_r \oplus {}^r T_q\}}{\partial v_r} \quad (\text{B.5})$$

This allows us to write the equations via the chain rule using the common differential operators defined in Equation B.7

$$\frac{\partial {}^m T_q}{\partial v_r} = \mathcal{J}_1({}^m T_{r-1} \oplus v_r, {}^r T_q) \mathcal{J}_2({}^m T_{r-1}, v_r) \quad (\text{B.6})$$

where

$$\begin{aligned} \mathcal{J}_1(x, y) &= \frac{\partial x \oplus y}{\partial x} \\ \mathcal{J}_2(x, y) &= \frac{\partial x \oplus y}{\partial y} \end{aligned} \quad (\text{B.7})$$

are the Jacobians of the composition operator $\oplus$ for arbitrary transformations x and y . For our implementation we have used a frobenius error metric ($d_M(a, b) = ||a - b||^2$) resulting in a non-linear least squares problem. With our VO system (Section 2.4.3) the state vector V is prohibitively large (a pose is created every 50 ms). We therefore re-write Equation B.3 into matrix form with a view to use conjugate gradients, making sure to exploit the sparsity of the interconnected graph to avoid memory issues at run-time

$$\delta {}^m T_q = \mathbf{h}_{m,q} \delta V \quad (\text{B.8})$$

where δV is a vector of small changes in V and $\mathbf{h}$ is a row-matrix where the k^{th} sub block ($m < k < q$) is given by Equation B.6 and zero for all k outside this range. We can encapsulate the first half of the cost function (Equation B.2) by seeking a change δV in V , such that the error between the constraints and the transforms (${}^m c_q - {}^m T_q = \delta^m c_q$) is zero, hence we seek:

$$\mathbf{h}_{m,q} \delta V = \delta^m c_q \quad (\text{B.9})$$

If we have n loop closure constraints we will have n such constraints to fulfill Equation B.9 yielding

$$\mathbf{H} \delta V = \delta C \quad (\text{B.10})$$

Including the second half of equation B.2 in a similar ilk, and pre-multiplying by $\begin{bmatrix} \mathbf{H}^T & I \end{bmatrix}$ we produce the normal equations

$$\begin{bmatrix} \mathbf{H}^T I \end{bmatrix} \begin{bmatrix} \mathbf{H} \\ I \end{bmatrix} \delta V = \begin{bmatrix} \mathbf{H}^T I \end{bmatrix} \begin{bmatrix} \delta C \\ O \end{bmatrix} \quad (\text{B.11})$$

We can now use conjugate gradients, exploiting the sparsity of $\begin{bmatrix} \mathbf{H}^T I \end{bmatrix} \begin{bmatrix} \mathbf{H} \\ I \end{bmatrix}$. Optimisation ceases when the perturbations in V become small.

Appendix C

3D Scan Matching via Iterative Closest Point

Iterative Closest Point (ICP) is a general algorithm that attempts to estimate the relative rigid transformation that will align two sets of points. It is commonly used within the mobile robotics industry as: it provides estimates that are robust to the typical noise in laser data, has a relatively low computational burden and requires no information about the correspondence between points in each point set. We use it to estimate the transformation between two 3D point clouds that have been generated by our research platform (as mentioned in Section 2.4.4).

ICP attempts to minimise the alignment error (as represented by an appropriately chosen error metric) between two point clouds through iteratively transforming one of the two clouds as shown in Figure C.1. On each iteration pairwise correspondences are re-assigned according to a heuristic; in the original algorithm of [12] this was the closest

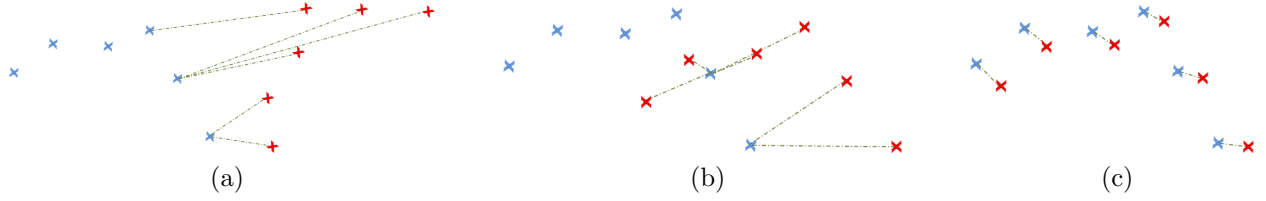


Figure C.1: Vanilla 2D ICP. Base cloud B is coloured blue, matching cloud M , red. Correspondences are taken to be the closest Euclidean point as indicated by the green dashed line. Each sub-figure represents the algorithm after an iteration, proceeding from left to right. During the first few ICP iterations, where there is gross miss-alignment, the correspondences are mostly incorrect as indicated by the one-to-many mappings from points in M to points in B in a) and b). As the algorithm converges the clouds overlap more and the true pairwise correspondences emerge as shown in c).

point, hence the algorithm’s name. The algorithm has remarkable convergence properties despite having no information on the correct pairwise correspondences.¹. That said, the convergence basin has been further improved with work such as histogram-based matching [15] that acts as a pre-conditioning stage for ICP. We have found that this increased basin is useful for our work, as we routinely match point clouds that are generated by the robot travelling in opposite directions (hence a yaw of roughly 180°). Here each pair of point clouds is formed from laser data that are gathered from around the same time as loop closures, which are dictated by our FAB-MAP algorithm [23] (see Section 2.4.4 for details).

The following canonical treatment, based on [92], is used to capture the plethora of publications on ICP. This analysis breaks ICP into six stages, of which we have included only the most important references (see [92] for a full list of citations). We pay particular attention to techniques aimed at increasing ICP robustness to noise in

¹If perfect correspondences are known then the closed form solution of [44] can be used to solve the transform in one iteration.

the point clouds. We then present the optimal solutions offered by [92] in terms of both speed and accuracy, and compare them with the Iterative Closest *Surface* of [15] and our approach.

C.1 ICP stages

Selection. - Subsets of point clouds from either the *base* (or model) cloud $B = [b_1, b_2, b_i \dots]^T$, *matching* (or data) cloud $M = [m_1, m_2, m_j \dots]^T$ or both are selected. Simple sampling techniques including complete, uniform and random sub-sampling chosen at either initialisation or on each iteration. Representative sampling has also been achieved through a variety of metrics such as in [20], commonly in the form of normal space sampling (whereby the distribution of point cloud surface normals are kept as uniform as possible). Additional information (such as point reflectivity or colouring from registered cameras) is also sometimes used to inform point selection. To allow a broad analysis [92] produce their point clouds from synthetic surface meshes. Whilst there was some attempt to add the effects of noisy and imperfect sensors on these surface meshes, the extent of the noise is not reported, nor is the degradation of techniques as the noise floor increases.

Matching - Finding the correct pairwise correspondence between one point cloud and another is non-trivial, especially when the points in the clouds do not sample at exactly the same locations, which is often the case with laser data. The original ICP algorithm of [12] states a simple closest point-to-point correspondence

$$c_j = \underset{i}{\operatorname{argmin}} \quad ||b_i - {}^B T_M m_j|| \quad (\text{C.1})$$

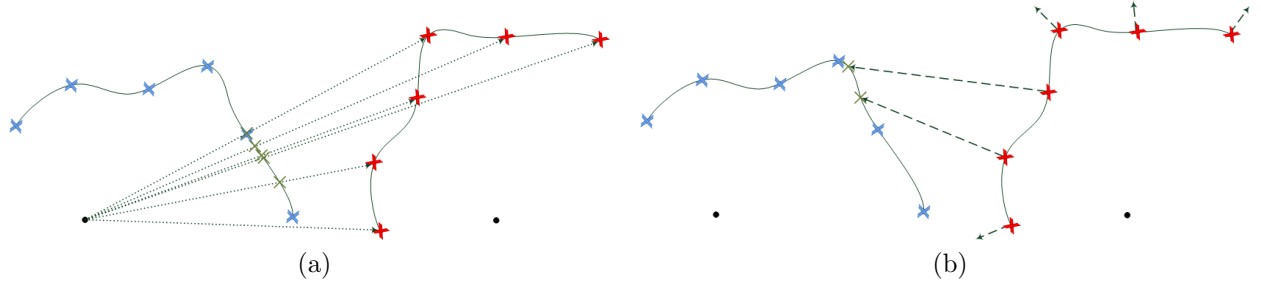


Figure C.2: Projection techniques work by projecting points from the centre of one cloud (often the origin of the emitted rays for a point cloud generated by a laser scanner or the focal point for those generated by a range camera) into the other. Base cloud B is coloured blue, matching cloud M , red. (a) Reverse calibration. A matching point m_i is projected from the centre of the base cloud and intersected with a surface constructed from the base cloud b'_i . Other projection techniques include finding the nearest base point b_i to b'_i . (b) Normal shooting. A matching point m_i is projected along its surface normal and intersected with a surface constructed from the base cloud b'_i . Although normal shooting is very effective in low noise conditions, it relies on the ability to capture correct point surface normals in the matching cloud which are difficult to calculate in the presence of a large amount of noise.

where c_j encodes the pairwise correspondence $C = [c_1, c_2, c_j \dots]^T$ between the point b_i and point m_j that is transformed through the current registration estimate ${}^B T_M$ (from base cloud B to matching cloud M) according to the smallest Euclidean norm (see Figure C.1). Although this remains the most robust and accurate technique for large point clouds over a wide range of scenarios, the operation of finding the nearest point is computationally intensive. Speed enhancements have resulted from efficient look-up techniques, such as a specifically modified k-d trees (applied to B), however this matching stage remains a bottleneck for ICP. Alternative techniques such as “normal shooting” [17] and reverse calibration (see Figure C.2) aim to increase the final accuracy of the estimate. Sometimes *compa-*

tibility metrics are also used, such as neighbouring colour or intensity similarities and the compatibility of normals. Often projection techniques are applied both from M to B and in the opposite direction to improve convergence and robustness to false matches.

Weighting - The weighting of each of the pairwise correspondences c_i affects the overall cost function that is to be minimised in the ICP algorithm. In the analysis of [92] they concentrated on how the compatibility metrics (from above) could be incorporated and how different weighting kernels could affect convergence rate.

Rejection - ICP robustness is increased through rejecting those c_i in gross error. This stage is particularly beneficial in scenarios where there are non-overlapping point clouds either because the clouds are being captured at slightly different locations in the environments, or because parts of a clouds are missing through occlusion. Convergence has been improved by rejecting points around the periphery of the clouds until final ICP iterations. Rejection techniques include: using a Euclidean distance threshold, rejecting the worst $n\%$ points, rejecting those c_i outside of a standard deviation in error, mesh boundary rejection and *consistency tests* to check that neighbouring correspondences are similar. Mesh boundary rejection is often used with techniques that rely on point normals, as it is difficult to estimate these normals in boundary areas due to a lack of surrounding points. In [15] they showed empirically that the adoption of a Lorentzian measurement model (Figure C.3) instead of an explicit rejection stage can be used to greatly increase the convergence basin and speed of ICP with only a small loss in accuracy. Intuitively, the Lorentzian distribution has much larger tails in the distribution,

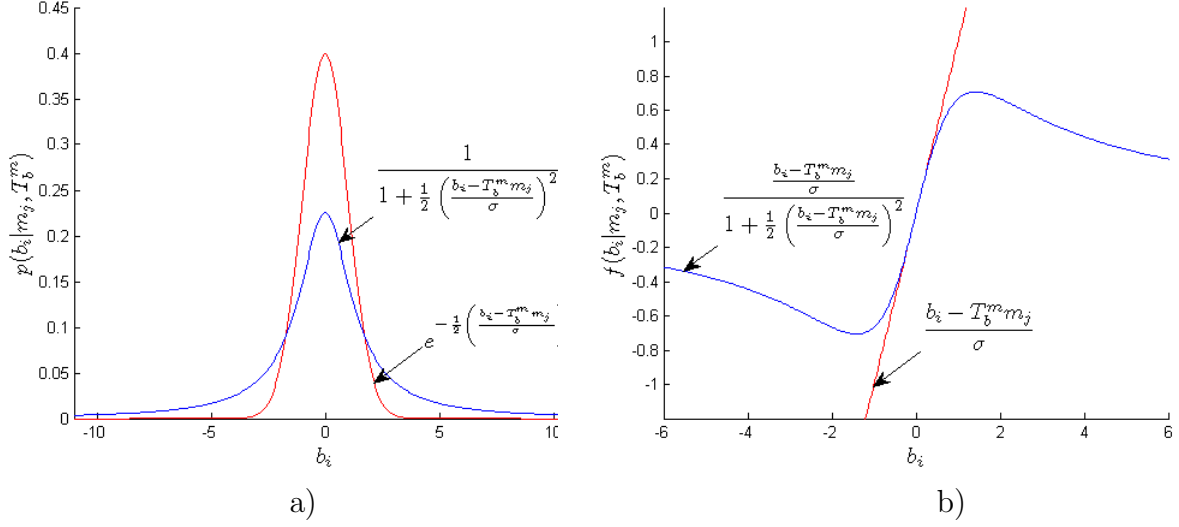


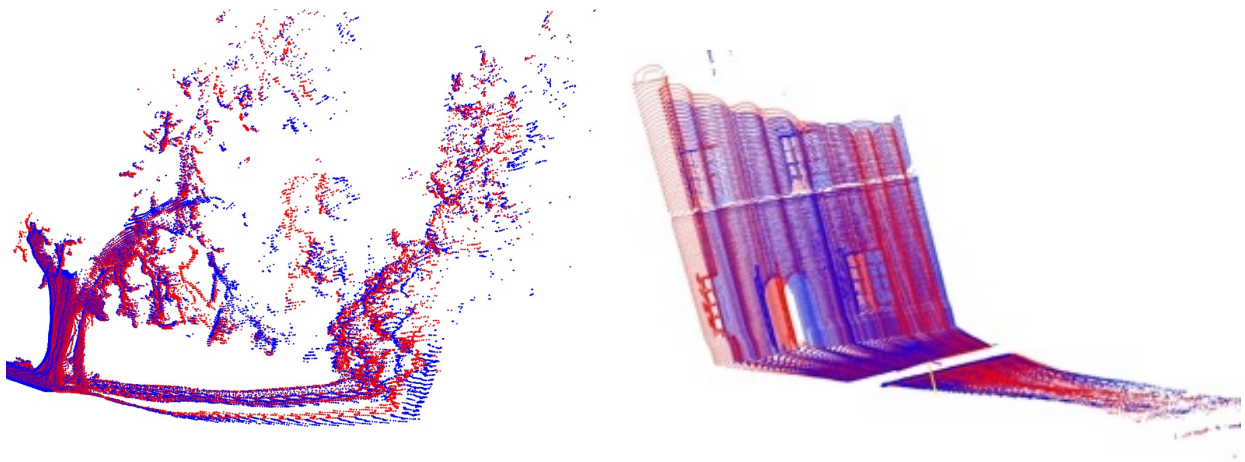
Figure C.3: Comparison of measurement models for ICP. (a) The original ICP algorithm [12] assumes a measurement error with Gaussian distribution around model prediction $T_b^m m_j$ in red, while [15] uses a Cauchy (Lorentzian) distribution, in blue. (b) The contribution of each point correspondence in the minimisation of the log likelihood. This is used to find the maximum likelihood estimation of the relative transformation between point clouds – see [86]. The Cauchy assumption (blue) provides a lower weighting at extreme distances than the Gaussian assumption (red). Poor correspondences (those with larger distances) therefore have less affect on each ICP iteration using the Cauchy assumption. Robustness to outliers and poor initial transformation estimates is therefore increased.

which allow correspondence outliers to be more accurately captured (see Figure C.3a).

Error Metric - To accurately capture the inter-cloud miss-alignment a cost function must be generated from an error metric over C . Originally the error metric was the Euclidean norm between points, but the shortest Euclidean distance from point-to-plane has also been used when surface meshes are available. Compatibility and consistency metrics (from the matching and rejection stages) are also

sometimes incorporated into the error metric. *Agreement metrics*, such as the dot product projection between the m_j surface normal and the corresponding matching direction (indicated by the green dashed line in Figure C.1) have also been substituted. All metrics permit quick optimisation techniques such as Gauss-Newton and Levenberg-Marquardt for increased robustness. Agreement metrics (where available) are preferable, as they help to avoid ICP failures such as the one in Figure C.4b. Here, the high planarity of the point clouds would have otherwise provided insufficient constraints to stop the clouds from sliding past each other on successive iterations on ICP.

Minimisation - Cost function minimisation is applied as the final stage of each of the iterations in the ICP algorithm. The initial transform for the ICP framework is commonly assumed to be the identity transform or hand-estimated. In our framework this initial estimate is supplied by a separate system such as FAB-MAP (Chapter 2.4.4) or alternatively the histogramming technique of [15] can be used to increase the convergence basin. At the cost of overshooting, the convergence rate can be increased through using optimisation techniques that are adapted to reduce the number of wasted ICP iterations. Higher accuracy has been obtained, through computationally-expensive simulated annealing and through the use of agreement methods across multiple ICP instantiations. Each instantiation for the latter is initialised with slightly perturbed initial transform estimations and the instantiation that results with, for example, the least median of squares is selected.



a) Good ICP alignment

b) Failed ICP alignment

Figure C.4: Scenes with rich geometry lead to excellent 6-DOF alignment but when presented with largely flat scenes ICP commonly converges to a local minima. (a) The boughs of the tree provide a well-defined minima and an excellent match between two point clouds (red and blue). (b) The facade of a building results in an incorrect alignment, explicable as a spatial-aliasing problem. The use of surface normals may help to reduce this problem. Alternatively the use of point clouds that have first been sub-sampled by our algorithms (developed in Chapters 6 and 7) may provide a similar role as normal space sampling, and therefore decrease the potential for poor alignment.

C.2 Comparison of Approaches

Table C.1 collates the suggestions from [92], [15] and the algorithm we have applied in Section 2.4.4. Figure C.4 depicts typical ICP results for our system.

ICP Stage	Iterative Closest Point[92] (Speed) (3D mesh)	Iterative Closest Point[92] (Accuracy) (3D mesh)	Iterative Closest Surface [15] (2D laser scan)	Our Approach (3D point cloud)
Selection	Random Sampling	largest normal distribution	all	all
Matching	single direction projection based	closest point	closest point (k-d tree accelerated)	closest point (k-d tree accelerated)
Weighting	constant weighting	compatibility of normals	lorentzian	lorentzian
Rejection	distance threshold	mesh boundary	non	non
Error Metric	sum squared distance point-to-plane	sum squared distance point-to-point	sum square normal to matching direction	sum squared distance point-to-point
Minimisation	standard iterative	standard iterative	standard iterative	standard iterative

Table C.1: 6 Stage ICP summary table. The suggestions in [92] that are aimed at producing a low computational burden are reported in the first column, and an algorithm that places primary importance on accuracy is reported in the second column. A summary of [15] and our implementation (as used in Section 2.4.4) is presented in columns three and four respectively. All techniques use standard optimisation techniques in the minimisation stage as alternatives tend to be either computationally-expensive, or incur a decrease in accuracy.

Appendix D

Properties of Gaussian distributions

A multivariate Gaussian distributed random variable $\mathbf{z}$ of size n can be fully specified by its second order statistics; its mean $\bar{\mathbf{z}}$ and its covariance Σ (as discussed in [13]):

$$\mathcal{N}(\mathbf{z}; \bar{\mathbf{z}}, \Sigma) \triangleq \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \exp \left(-\frac{1}{2} (\mathbf{z} - \bar{\mathbf{z}})^{\mathbf{T}} \Sigma^{-1} (\mathbf{z} - \bar{\mathbf{z}}) \right) \quad (\text{D.1})$$

where

$$\bar{\mathbf{z}} = \mathbb{E}[\mathbf{z}] \quad (\text{D.2})$$

$$\Sigma = \mathbb{V}[\mathbf{z}] = \mathbb{E}[(\mathbf{z} - \bar{\mathbf{z}})^T (\mathbf{z} - \bar{\mathbf{z}})] \quad (\text{D.3})$$

We have used $\mathbb{E}[\mathbf{z}]$ to indicate the expected value of the random variable $\mathbf{z}$

$$\mathbb{E}[\mathbf{z}] = \int \mathbf{z} p(\mathbf{z}) d\mathbf{z} \quad (\text{D.4})$$

D.1 Joint Gaussian distributed variables

We may freely partition a multivariate random variable $\mathbf{z}$ into $\mathbf{x}$ and $\mathbf{y}$

$$\mathbf{z} \triangleq \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \quad (\text{D.5})$$

When the two sets of variables $\mathbf{x}$ and $\mathbf{y}$ are joint Gaussian distributed we may also define:

$$p(\mathbf{x}, \mathbf{y}) \triangleq \mathcal{N} \left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}; \begin{bmatrix} \bar{\mathbf{x}} \\ \bar{\mathbf{y}} \end{bmatrix}, \begin{bmatrix} \Sigma_{\mathbf{xx}} & \Sigma_{\mathbf{xy}} \\ \Sigma_{\mathbf{xy}}^T & \Sigma_{\mathbf{yy}} \end{bmatrix} \right) \quad (\text{D.6})$$

Covariance matrix subscripts denote either block variances of the corresponding random variable ($\Sigma_{\mathbf{xx}}$ and $\Sigma_{\mathbf{yy}}$), or the block covariance between random variables ($\Sigma_{\mathbf{xy}}$ and $\Sigma_{\mathbf{yx}} = \Sigma_{\mathbf{xy}}^T$ due to symmetry). We can confirm that the mean and covariance of Equation D.6 follow the definitions of Equations D.2 and D.3 through use of the *marginalisation property* (or alternatively the *consistency property*) of a multivariate Gaussian distribution. That is the marginals $p(\mathbf{x})$ and $p(\mathbf{y})$ satisfy $\mathbf{x} \sim \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \Sigma_{\mathbf{xx}})$ and $\mathbf{y} \sim \mathcal{N}(\mathbf{y}; \bar{\mathbf{y}}, \Sigma_{\mathbf{yy}})$ and hence that examination of the larger set of variables $\mathbf{z}$ (with distribution given by Equation D.6) does not change the distributions of the smaller sets. These marginals are calculated through integrating out the relevant random variable, for example:

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{y}) d\mathbf{y} \quad (\text{D.7})$$

Following the approach in [13] we introduce the corresponding block precision matrix $\Lambda = \Sigma^{-1}$ to expedient this integration (the correspondance between elements in Λ and Σ are well known and may be found through the partition matrix formula - see [86] for example).

$$\Sigma = \begin{bmatrix} \Sigma_{\mathbf{xx}} & \Sigma_{\mathbf{xy}} \\ \Sigma_{\mathbf{xy}}^T & \Sigma_{\mathbf{yy}} \end{bmatrix} = \Lambda^{-1} = \begin{bmatrix} \Lambda_{\mathbf{xx}} & \Lambda_{\mathbf{xy}} \\ \Lambda_{\mathbf{xy}}^T & \Lambda_{\mathbf{yy}} \end{bmatrix}^{-1} \quad (\text{D.8})$$

This formulation permits us to expand the quadratic form of the exponent in the definition of a multivariate Gaussian distribution (Equation D.1) in terms of its block elements and group in according to $\mathbf{y}$. Noting that this expansion is of quadratic form in $\mathbf{y}$ the integration is therefore that of integrating an unnormalised Gaussian distribution and the result will therefore be the reciprical to the normalising coefficient. We are left with a Gaussian distribution dependent only on $\mathbf{x} \sim \mathcal{N}(\mathbf{x}; \mu_{\mathbf{x}}, \Sigma_{\mathbf{x}})$, of which we may compare linear and quadratic terms

$$\mathbf{x}^T \Sigma_{\mathbf{x}}^{-1} \mu_{\mathbf{x}} = \mathbf{x}^T (\Lambda_{\mathbf{xx}} - \Lambda_{\mathbf{xy}} \Lambda_{\mathbf{yy}}^{-1} \Lambda_{\mathbf{xy}}^T) \bar{\mathbf{x}} \quad (\text{D.9})$$

$$\mathbf{x}^T \Sigma_{\mathbf{x}}^{-1} \mathbf{x} = \mathbf{x}^T (\Lambda_{\mathbf{xx}} - \Lambda_{\mathbf{xy}} \Lambda_{\mathbf{yy}}^{-1} \Lambda_{\mathbf{xy}}^T) \mathbf{x} \quad (\text{D.10})$$

The partition matrix formula provides us the identity $\Lambda_{\mathbf{xx}} - \Lambda_{\mathbf{xy}} \Lambda_{\mathbf{yy}}^{-1} \Lambda_{\mathbf{xy}}^T \triangleq \Sigma_{\mathbf{xx}}^{-1}$ and therefore $\Sigma_{\mathbf{x}} = \Sigma_{\mathbf{xx}}$ and $\mu_{\mathbf{x}} = \bar{\mathbf{x}}$ as desired.

For our work we often wish to form the conditional probability of $\mathbf{x}$ given $\mathbf{y}$, $p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{y})}$. By following the same considerations as above we may group block elements in terms of $\mathbf{x}$ and hence the conditional distribution will also be Gaussian $p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}; \mu_{\mathbf{x}|\mathbf{y}}, \Sigma_{\mathbf{x}|\mathbf{y}})$. Comparing linear and quadratic terms reveals

$$\mathbf{x}^T \Sigma_{\mathbf{x}|\mathbf{y}}^{-1} \mu_{\mathbf{x}|\mathbf{y}} = \mathbf{x}^T (\Lambda_{\mathbf{xx}} \bar{\mathbf{x}} - \Lambda_{\mathbf{xy}} (\mathbf{y} - \bar{\mathbf{y}})) \quad (\text{D.11})$$

$$\mathbf{x}^T \Sigma_{\mathbf{x}|\mathbf{y}}^{-1} \mathbf{x} = \mathbf{x}^T \Lambda_{\mathbf{xx}} \mathbf{x} \quad (\text{D.12})$$

Therefore, collating terms

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}; \Lambda_{\mathbf{xx}}^{-1} (\Lambda_{\mathbf{xx}} \bar{\mathbf{x}} - \Lambda_{\mathbf{xy}} (\mathbf{y} - \bar{\mathbf{y}})), \Lambda_{\mathbf{xx}}^{-1}) \quad (\text{D.13})$$

Whilst this form of the conditional distribution is useful when we consider linear Gaussian models (Section 4.2.1) we are generally concerned with joint Gaussian distributed variables in terms of covariance matrices. The partitioned matrix formula permits us to write

$$\Lambda_{\mathbf{xx}} = (\Sigma_{\mathbf{xx}} - \Sigma_{\mathbf{xy}} \Sigma_{\mathbf{yy}}^{-1} \Sigma_{\mathbf{xy}}^T)^{-1} \quad (\text{D.14})$$

$$\Lambda_{\mathbf{xy}} = -\Lambda_{\mathbf{xx}} \Sigma_{\mathbf{xy}} \Sigma_{\mathbf{yy}}^{-1} \quad (\text{D.15})$$

Where $\Lambda_{\mathbf{xx}}$ is the inverse of the Schur complement of Σ with respect to $\Sigma_{\mathbf{yy}}$. We can therefore use these two identities to rephrase Equation D.13 into a form which we

shall use throughout our discussions on Gaussian Process Regression in Section 4.4

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}} + \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{yy}}^{-1}(\mathbf{y} - \bar{\mathbf{y}}), \Sigma_{\mathbf{xx}} - \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{yy}}^{-1}\Sigma_{\mathbf{xy}}^T) \quad (\text{D.16})$$

Bibliography

- [1] W. Aaron, F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin. MAPS: Multiresolution Adaptive Parameterization of Surfaces. In *SIGGRAPH*, 1998.
- [2] M. Abramowitz and I.A. Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. Dover publications, 1965.
- [3] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C.T. Silva. Point set surfaces. In *Proceedings of the Conference on Visualization'01*, pages 21–28. IEEE Computer Society, 2001.
- [4] M. Alexa, M. Gross, M. Pauly, H. Pfister, M. Stamminger, and M. Zwicker. Point-based computer graphics. In *ACM SIGGRAPH 2004 Course Notes*, pages 7–es. ACM, 2004.
- [5] N. Amenta, M. Bern, and M. Kamvysselis. A new Voronoi-based surface reconstruction algorithm. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 415–421. ACM, 1998.

- [6] N. Amenta, S. Choi, and R.K. Kolluri. The power crust. In *Proceedings of the sixth ACM symposium on Solid modeling and applications*, pages 249–266. ACM, 2001.
- [7] C.F. Ansley and R. Kohn. Estimation, filtering, and smoothing in state space models with incompletely specified initial conditions. *The Annals of Statistics*, 13(4):1286–1316, 1985.
- [8] C.L. Bajaj, F. Bernardini, and G. Xu. Automatic reconstruction of surfaces and scalar fields from 3d scans. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 109–118. ACM, 1995.
- [9] J. Bares, M Hebert, T. Kanade, E. Krotkov, T. Mitchell, R. Simmons, and W. R. L. Whittaker. Ambler: An autonomous rover for planetary exploration. *IEEE Computer*, 22(1):18–26, 1989.
- [10] M. Bayes and M. Price. An essay towards solving a problem in the doctrine of chances. by the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFRS. *Philosophical Transactions*, 53:370, 1763.
- [11] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *Visualization and Computer Graphics, IEEE Transactions on*, 5(4):349–359, 1999.
- [12] P.J. Besl and H.D. McKay. A method for registration of 3D shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256, 1992.
- [13] C.M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

- [14] B.J.N. Blight and L. Ott. A Bayesian approach to model inadequacy for polynomial regression. *Biometrika*, 62(1):79, 1975.
- [15] M. Bosse and R. Zlot. Map matching and data association for large-scale two-dimensional laser scan-based SLAM. In *International Journal of Robotics Research*, volume 27, pages 667–691. SAGE Publications, June 2008.
- [16] J.C. Carr, R.K. Beatson, J.B. Cherrie, T.J. Mitchell, W.R. Fright, B.C. McCallum, and T.R. Evans. Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 67–76. ACM, 2001.
- [17] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *1991 IEEE International Conference on Robotics and Automation, 1991. Proceedings.*, pages 2724–2729, 1991.
- [18] P. Cignoni, C. Montani, and R. Scopigno. A comparison of mesh simplification algorithms. *Computers & Graphics*, 22(1):37–54, 1998.
- [19] D.M. Cole and P.M. Newman. Context and feature sensitive re-sampling from discrete surface measurements. In *Robotics: Science and Systems (RSS)*. Citeseer, 2007.
- [20] D.M. Cole and P.M. Newman. Context and feature sensitive re-sampling from discrete surface measurements. In *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007. RSS.

- [21] L. Csató and M. Oppér. Sparse on-line Gaussian processes. *Neural Computation*, 14(3):641–668, 2002.
- [22] G. Cuccuru, E. Gobbetti, F. Marton, R. Pajarola, and R. Pintus. Fast low-memory streaming MLS reconstruction of point-sampled surfaces. In *Proceedings of Graphics Interface 2009*, pages 15–22. Canadian Information Processing Society, 2009.
- [23] M. Cummins and P.M. Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.
- [24] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312. ACM, 1996.
- [25] M.P. Deisenroth, C.E. Rasmussen, and J. Peters. Gaussian process dynamic programming. *Neurocomputing*, 72(7-9):1508–1524, 2009.
- [26] M. Desbrun, M. Meyer, P. Schröder, and A.H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 317–324. ACM Press/Addison-Wesley Publishing Co., 1999.
- [27] J.J. Dongara, C.B. Moler, J.R. Bunch, and G.W. Stewart. *LINPACK users' Guide*. SIAM, 1979.

- [28] M. El-Beltagy and W. Wright. Gaussian processes for model fusion. *Artificial Neural Networks (ICANN 2001)*, 2130/2001:376–383, 2001.
- [29] U. Frese, P. Larsson, and T. Duckett. A multilevel relaxation algorithm for simultaneous localization and mapping. *Robotics, IEEE Transactions on*, 21(2):196–207, 2005.
- [30] C. Früh and A. Zakhor. Constructing 3d city models by merging aerial and ground views. *IEEE Comput. Graph. Appl.*, 23(6):52–61, 2003.
- [31] P.E. Gill, G.H. Golub, W. Murray, and M.A. Saunders. Methods for modifying matrix factorizations. *Mathematics of Computation*, 28(126):505–535, 1974.
- [32] M. Gopiy, S. Krishnan, and C.T. Silva. Surface reconstruction based on lower dimensional localized delaunay triangulation. In *Computer Graphics Forum*, volume 19, pages 467–478. Wiley Online Library, 2000.
- [33] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Proc. of Robotics: Science and Systems (RSS)*, 2007.
- [34] G. Guennebaud and M. Gross. Algebraic point set surfaces. In *ACM SIGGRAPH 2007 papers*, pages 23–es. ACM, 2007.
- [35] R. Hadsell, J.A. Bagnell, D. Huber, and M. Hebert. Non-Stationary Space-Carving Kernels for Accurate Rough Terrain Estimation. *International Journal for Robotics Research (IJRR)*, 29(8):981–996, July 2010.

- [36] D. Hähnel, W. Burgard, and S. Thrun. Learning compact 3D models of indoor and outdoor environment with a mobile robot. *Robotics and Autonomous Systems*, 44(1):15–27, 2003.
- [37] A. Harrison and P. Newman. Ticsync: Knowing when things happened. In *Proc. IEEE International Conference on Robotics and Automation (ICRA2011)*, Shanghai, China, May 2011. 05.
- [38] P. Hebert, D. Laurendeau, and D. Poussart. Scene reconstruction and description: geometric primitive extraction from multiple viewed scattered data. In *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93., 1993 IEEE Computer Society Conference on*, pages 286–292. IEEE, 1993.
- [39] P.S. Heckbert and M. Garland. *Survey of polygonal surface simplification algorithms*. Citeseer, 1997.
- [40] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using depth cameras for dense 3d modeling of indoor environments. In *the 12th International Symposium on Experimental Robotics (ISER)*, 2010.
- [41] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise smooth surface reconstruction. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 295–302. ACM, 1994.

- [42] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics Association for Computing Machinery*, 26:71–71, 1992.
- [43] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 19–26. ACM New York, NY, USA, 1993.
- [44] B.K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.
- [45] A.E. Johnson. *Spin-Images: A Representation for 3D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997.
- [46] A. Kalaiah and A. Varshney. Differential point rendering. In *In Proceedings of the 12th Eurographics Workshop on Rendering*. Citeseer, 2001.
- [47] M. Kazhdan. Reconstruction of solid models from oriented point sets. In *Proceedings of the third Eurographics symposium on Geometry processing*, page 73. Eurographics Association, 2005.
- [48] J. Ko, B. Stewart, D. Fox, K. Konolige, and B. Limketkai. A practical, decision-theoretic approach to multi-robot mapping and exploration. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 4, pages 3232–3238. IEEE, 2003.

- [49] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research*, 9:235–284, 2008.
- [50] V. Krishnamurthy and M. Levoy. Fitting smooth surfaces to dense polygon meshes. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 313–324. ACM, 1996.
- [51] R. Kummerle, B. Stederand, C. Dornhege, A. Kleiner, G. Grisetti, and W. Burgard. Large Scale Graph-based SLAM using Aerial Images as Prior Information. In *Proceedings of Robotics: Science and Systems*. Citeseer, 2008.
- [52] T. Lang, C. Plagemann, and W. Burgard. Adaptive non-stationary kernel regression for terrain modeling. In *Robotics: Science and Systems (RSS)*, 2007.
- [53] I.K. Lee. Curve reconstruction from unorganized points. *Computer Aided Geometric Design*, 17(2):161–177, 2000.
- [54] D. Levin. Mesh-independent surface interpolation. *Geometric modeling for scientific visualization*, 3:37–49, 2003.
- [55] M. Levoy and T. Whitted. The use of points as a display primitive. Technical report, University of North Carolina at Chapel Hill, 1985.
- [56] L. Linsen. Point cloud representation. Technical report, University of Karlsruhe, 2001.
- [57] D.J.C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge University Press, 2003.

- [58] J. Manson, G. Petrova, and S. Schaefer. Streaming surface reconstruction using wavelets. In *Computer Graphics Forum*, volume 27, pages 1411–1420. Wiley Online Library, 2008.
- [59] M. C. Martin and H. Moravec. Robot evidence grids. Technical Report CMU-RI-TR-96-06, Robotics Institute, 1996.
- [60] G. Matheron. The intrinsic random functions and their applications. *Advances in Applied Probability*, 5(3):439–468, 1973.
- [61] The Mathworks. MATLAB 2008a. Natick, MA, 2008.
- [62] D.H. McLain. Drawing contours from arbitrary data points. *The Computer Journal*, 17(4):318, 1974.
- [63] B. Mederos, N. Amenta, L. Velho, and L.H. de Figueiredo. Surface reconstruction from noisy point clouds. In *Proceedings of the third Eurographics symposium on Geometry processing*, pages 53–es. Eurographics Association, 2005.
- [64] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid. A constant time efficient stereo slam system. In *BMVC*, 2009.
- [65] A. Napier, S. Sibley, and P. Newman. Real-time bounded-error pose estimation for road vehicles using vision. In *13th Intl. IEEE Conference on Intelligent Transportation Systems (ITSC2010?)*, pages pp. 1141–1146, Funchal, Portugal, September 2010.

- [66] R.M. Neal. Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. Technical Report 9702, Dept. of Statistics, University of Toronto, 1997.
- [67] P. Newman, G. Sibley, M. Cummins, M. Smith, A. Harrison, C. Mei, and I. Reid. Toward life long mapping. In *Proc. of the Int. Symposium of Robotics Research (ISRR)*, Lucerne, Switzerland, August 2009.
- [68] P.M. Newman. MOOS - a mission oriented operating suite. Technical Report Technical Report OE2003-07, MIT Department of Ocean Engineering, <http://www.robots.ox.ac.uk/~pnewman/TheMOOS>, 2003.
- [69] P.M. Newman, G. Sibley, M. Smith, M. Cummins, A.R. Harrison, C. Mei, I. Posner, R. Shade, D. Schroeter, D.M. Cole, and I Reid. Navigating, recognising and describing urban spaces with vision and laser. *IJRR*, 28(11-12):1406–1433, November 2009.
- [70] D. Nguyen-Tuong and J. Peters. Using Model Knowledge for Learning Inverse Dynamics. In *IEEE International Conference on Robotics and Automation*, 2010.
- [71] D. Nguyen-Tuong, M. Seeger, and J. Peters. Computed torque control with non-parametric regression models. In *American Control Conference, 2008*, pages 212–217. IEEE, 2008.
- [72] A. O’Hagan. Monte Carlo is fundamentally unsound. *The Statistician*, 36(2):247–249, 1987.

- [73] A. O’Hagan. Bayes-Hermite quadrature. *Journal of Statistical Planning and Inference*, 29(3):245–260, 1991.
- [74] A. O’Hagan and J.F.C. Kingman. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, 40:1–42, 1978.
- [75] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.P. Seidel. Multi-level partition of unity implicits. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 463–470. ACM, 2003.
- [76] E. Olson, J. Leonard, and S. Teller. Fast iterative alignment of pose graphs with poor initial estimates. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2262–2269. Ieee, 2006.
- [77] M. Osborne. *Bayesian Gaussian Processes for Sequential Prediction, Optimisation and Quadrature*. PhD thesis, University of Oxford, 2010.
- [78] M.A. Osborne, S.J. Roberts, A. Rogers, S.D. Ramchurn, and NR Jennings. Towards real-time information processing of sensor network data using computationally efficient multi-output Gaussian processes. In *Proceedings of the 7th international conference on Information processing in sensor networks*, pages 109–120. IEEE Computer Society Washington, DC, USA, 2008.
- [79] M. Pauly, M. Gross, and L.P. Kobbelt. Efficient simplification of point-sampled surfaces. In *Visualization, 2002. VIS 2002. IEEE*, pages 163–170. IEEE, 2002.
- [80] P. Pfaff and W. Burgard. An efficient extension of elevation maps for outdoor terrain mapping. In *Intl. Conf. on Field and Service Robotics (FSR)*, 2005.

- [81] H. Pfister, M. Zwicker, J. Van Baar, and M. Gross. Surfels: Surface elements as rendering primitives. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 335–342. ACM Press/Addison-Wesley Publishing Co., 2000.
- [82] L. Piegl. On NURBS: a survey. *computer Graphics and Applications, IEEE*, 11(1):55–71, 1991.
- [83] L.A. Piegl and W. Tiller. *The NURBS book*. Springer Verlag, 1997.
- [84] C. Plagemann, K. Kersting, P. Pfaff, and W. Burgard. Gaussian beam processes: A nonparametric Bayesian measurement model for range finders. In *Robotics: Science and Systems (RSS)*, 2007.
- [85] C. Plagemann, S. Mischke, S. Prentice, K. Kersting, N. Roy, and W. Burgard. Learning predictive terrain models for legged robot locomotion. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3545–3552, 2008.
- [86] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical recipes in C++*. Cambridge University Press, 3rd edition, 2007.
- [87] J. Quiñonero-Candela and C.E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *J. Mach. Learn. Res.*, 6:1939–1959, 2005.
- [88] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

- [89] C.E. Rasmussen and Z. Ghahramani. Bayesian Monte Carlo. In *Advances in neural information processing systems*, pages 505–512. Citeseer, 2003.
- [90] I. Rekleitis, J.L. Bedwani, and E. Dupuis. Over-the-horizon, autonomous navigation for planetary exploration. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 2248–2255. IEEE, 2007.
- [91] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1508–1515 Vol. 2, Oct. 2005.
- [92] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proc. 3DIM*, pages 145–152, 2001.
- [93] B. Scholkopf and A.J. Smola. *Learning with kernels*. the MIT Press, 2002.
- [94] M. Seeger. Low rank updates for the Cholesky decomposition. Technical report, University of California at Berkeley, 2008.
- [95] M. Seeger, C.K.I. Williams, and N.D. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In *Workshop on AI and Statistics*, volume 9, 2003.
- [96] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings*, pages 1470–1477, 2003.

- [97] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P.M. Newman. The New College vision and laser data set. *International Journal for Robotics Research (IJRR)*, 28(5):595–599, May 2009.
- [98] M. Smith, I. Posner, and P. Newman. Efficient non-parametric surface representations using active sampling for push broom laser data. In *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, 2010.
- [99] M. Smith, I. Posner, and P. Newman. Generating implicit surfaces from lidar data. In *Towards Autonomous Robotic Systems*, Plymouth, UK, 2010.
- [100] M. Smith, I. Posner, and P. Newman. Adaptive compression for 3D laser data. *The International Journal of Robotics Research*, 30(7):914 – 935, June 2011.
- [101] M. Smith, I. Posner, and P. M. Newman. Adaptive data compression for robot perception. In *Proceedings of The International Joint Conference on Artificial Intelligence (IJCAI)*, Barcelona, Spain, July 2011.
- [102] M. Soucy and D. Laurendeau. Surface modeling from dynamic integration of multiple range views. In *Pattern Recognition, 1992. Vol. I. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference on*, pages 449–452. IEEE, 1992.
- [103] M. L. Stein. *Interpolating Spatial Data*. Springer-Verlag, 1999.
- [104] G. Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358. ACM, 1995.

- [105] S. Thrun and M. Montemerlo. The graph SLAM algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research*, 25(5-6):403, 2006.
- [106] R. Triebel, W. Burgard, and F. Dellaert. Using hierarchical EM to extract planes from 3d range scans. In *Int. Conf. Robotics and Automation (ICRA)*, 2005.
- [107] R. Triebel, P. Pfaff, and W. Burgard. Multi-level surface maps for outdoor terrain mapping and loop closing. In *Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2006.
- [108] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 311–318. ACM, 1994.
- [109] G.E. Uhlenbeck and L.S. Ornstein. On the theory of the Brownian motion. *Physical Review*, 36(5):823, 1930.
- [110] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte. Gaussian process modeling of large-scale terrain. *Journal of Field Robotics*, 26(10):812–840, 2009.
- [111] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte. Large-scale terrain modeling from multiple sensors with dependent Gaussian processes. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1215–1221. IEEE, 2010.
- [112] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte. Non-stationary dependent Gaussian processes for data fusion in large-scale terrain modeling. In

Robotics and Automation (ICRA), 2011 IEEE International Conference on, pages 1875–1882, may 2011.

- [113] M.D. Wheeler, Y. Sato, and K. Ikeuchi. Consensus surfaces for modeling 3d objects from multiple range images. *Modeling from Reality*, 640:77–92, 2002.
- [114] M.A. Woodbury. Inverting modified matrices. *Memorandum report*, 42:106, 1950.
- [115] C. Ye and J. Borenstein. A new terrain mapping method for mobile robots obstacle negotiation. In *Proc. the UGV Technology Conference at the 2003 SPIE AeroSense Symposium*, pages 21–25. Citeseer, 2003.