Adaptive compression for 3D laser data

Mike Smith, Ingmar Posner and Paul Newman

The International Journal of Robotics Research 2011 30: 914
DOI: 10.1177/0278364911403019

The International Journal of
Robotics Research
30(7) 914–935
© The Author(s) 2011
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0278364911403019
ijr.sagepub.com

Adaptive compression for 3D laser data

Mike Smith, Ingmar Posner and Paul Newman

Abstract

This paper concerns the creation of efficient surface representations from laser point clouds created by a push broom laser system. We produce a continuous, implicit, non-parametric and non-stationary representation with an update time that is constant. This allows us to form predictions of the underlying workspace surfaces at arbitrary locations and densities. The algorithm places no restriction on the complexity of the surfaces and automatically prunes redundant data via an information theoretic criterion. This criterion makes the use of Gaussian Process Regression a natural choice. We adopt a formulation which handles the typical non-functional relation between XY location and elevation allowing us to map arbitrary environments. Results are presented that use real and synthetic data to analyse the trade-off between compression rate and reconstruction error. We attain decimation factors in excess of two orders of magnitude without significant degradation in fidelity.

Keywords

3D mapping, range sensing, Gaussian Process, mobile robotics, online representation, non-stationary

1. Introduction and motivation

Three-dimensional (3D) point clouds gathered from outdoor scenes are now ubiquitous. Whilst visually appealing when rendered, they have limitations, two of which are addressed in this paper. First, they are discrete samples and do not readily admit access to the continuous nature of the workspace's surfaces. Second, they are inefficient in the sense of containing samples which, given neighbouring samples and reasonable smoothness assumptions, are redundant and convey little additional information about the underlying scene geometry.

Our motivation lies in the value of obtaining richer workspace descriptions generated by a mobile robot. We restrict our attention to point clouds generated from the simplest and cheapest of 3D data acquisition systems: the push broom laser. Here, as shown in Figure 1, a single 2D laser scanner is pushed through the workspace and a point cloud is formed by aggregation. We describe a non-parametric method 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.

At the heart of our method is the application of a Gaussian process (GP) to select support from a finite, time varying region. This GP is used to generate a predictive distribution of individual range measurements in constant time regardless of the quantity of the collected laser data. By evaluating the similarity between the distribution of the measurement (using a sensor model) and its corresponding GP-predicted distribution, we arrive at a criterion for keeping or dropping individual samples. Here we draw inspiration from work in sparse GPs (Csató and Opper 2002; Seeger et al. 2003; Quiñonero-Candela and Rasmussen 2005). We refer to this process as *active sampling*: the GP framework actively guides us in deciding which of many input measurements should be retained and used later for surface reconstructions. To reduce algorithm complexity our approach reuses computations whenever possible.

Our approach admits the representation of every (x, y, z) point that lies on a sampled workspace surface. A particular advantage of our formulation is the ability to represent

Oxford University, Mobile Robotics Group, Oxford, UK

Corresponding author:
Mike Smith, Oxford University, Mobile Robotics Group, Parks Road Oxford OX1 3PJ, UK
Email: mike@robots.ox.ac.uk

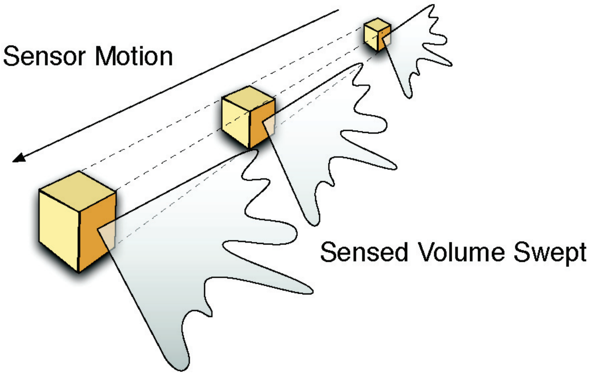


Fig. 1. The geometry of a push broom laser. A single 2D sensor is pushed through the environment, sweeping out a sensed volume.

common structures such as walls, which require multiple elevation values to be associated with the same position on the x–y plane (Figure 2). Technically, this multivalued *relation* is *non-functional*, and we use this term to distinguish it from that of a single-valued, *well-formed* function.

Above and beyond our previous work (Smith et al. 2010a), this paper also explores a non-stationary variant of our original approach, based on Bayesian Monte Carlo. Here a bank of stationary GP regressors are combined with complexity linear in the size of the bank.

2. Overview

In this section we provide a map of how the various components of our algorithm fit together. 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 emphasize 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.

Specifically, we pose the problem as that of regression using a GP. Consider a 2D laser scanner being dragged through space along a known curve which is parameterized by time, and a beam parameterized by a single angle. Using these parameters we define, in Section 4, a *sensor configuration space*. We use a GP to predict range measurements for arbitrary sensor configurations based (‘*conditioned*”) on real laser measurements. These predictions can also be transformed into a more traditional Euclidean x, y, z representation using the machinery in Equations (3) and (4). This parameterization also means that we can perform our regression independently of the task of trajectory estimation.

We detail our framework in Section 5 starting with a summary of the aspects of the GP literature that are used in our approach. GPs afford non-parametric, probabilistic regression that we use to provide range predictions via Equation (8) and uncertainties via Equation (9). However, a well-known drawback of GPs is that they scale as $\mathcal{O}(N^3)$ in the number of observations used to condition the predictions. This motivates Section 5.1 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. Another property of GPs is that in areas bereft of data, predictions tend towards the GP’s mean function. It is important that we do not use the common 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 fitting with only a few terms works effectively for our problem. We allow our regression to remain non-parametric by wrapping the fitting into the overall GP regression in Section (5.2).

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 (5.3) 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. To account for the uncertainty in our predictions as well as the range estimate we take the Kullback–Leibler (KL) divergence (Equation (18)) between the predictive distribution and the distribution provided by a sensor measurement model. 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.

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. When a new measurement needs to be added to the active window we update our beliefs using the measurement rather than recomputing from all measurements in the active window. This approach is general in that it can be applied to any *stationary* GP, that is, a GP with a covariance function that does not change across the dataset. We also introduce the notion of *introspection* in Section (5.3) whereby we allow *post hoc* conditioning from laser measurements previously thought to be uninformative. We show how this approach of reconsidering earlier decisions is important in the vicinity of abrupt range discontinuities.

Finally in Section 9, we demonstrate how our approach can be extended to use a *non-stationary* GP. In essence we combine multiple instantiations of our stationary GPs by weighting each of their predictions according to how well each regressor fits the current data. Predictions are therefore automatically highly weighted by GPs that perform well for rough surfaces, in rough areas of the dataset; and by GPs aimed at smooth surfaces, in smoother areas of the dataset. Data fit is determined by two alternative approximations of predictive Bayesian quadrature. These approximations produce an algorithm that scales linearly with the number of GP instantiations and is trivial to parallelize. To exploit the increased model flexibility we also make simple modifications to the covariance function of Section 5, and other possible modifications are discussed in Section 9.1. On comparing stationary results (Section 8) with non-stationary results (Section 10) we find, for a given number of retained laser measurements, that the non-stationary extension provides marginally better surface prediction accuracy. The non-stationary extension also provides an indication of surface properties which could be useful for other processes.

3. Related work

Within the robotics community, 3D surface reconstruction from sparsely sampled environments is commonly motivated by the need to model an unknown terrain or to provide an accurate reconstruction of the workspace across scales. The latter is commonly achieved by meshing techniques (see, for example, Früh and Zakhor (2003) or Hoppe et al. (1993)) whereby every datum of a point cloud forms the vertex of a polygon. Often such techniques are augmented by a mesh decimation step which aims to obtain a good representation of the environment, while at the same time reducing the amount of data to be stored. Commonly, mesh creation as well as the reduction in the number of vertices is 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.

An alternative approach considers a parametric representation based on geometric primitives (Hähnel et al. 2003; Triebel et al. 2005). 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.

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 work in 3D reconstruction techniques trace their origin to this domain. Common approaches include discrete 2D, 2.5D and 3D methods such as those of Martin and Moravec (1996), Bares et al. (1989), Pfaff and Burgard (2005), and Triebel et al. (2006), 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.

In recent years there has been a further trend in the terrain modelling community to employ non-parametric continuous models of surfaces. 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 Lang et al. (2007), Plagemann et al. (2008), Vasudevan et al. (2009), and Hadsell et al. (2010), although they were first introduced to the geostatistics field under the name *kriging* many years previously by Matheron (1973) amongst others. While these approaches differ in their choice of (non-stationary) covariance function and GP sparsification method, they adopt the same parameterization 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. Hadsell et al. (2010) exploit this limitation by constraining their surface regression with visibility information of the sensor. While this well-formed functional mapping is effective for terrain modelling it precludes application in the full 3D mapping case (Figure 2). Our work aims to overcome this limitation by introducing a more natural parameterization of the surface reconstruction problem, as well as providing a principled method to sub-sample the data.

There has been much research into sparsification for GP regression. An overview is provided in Quiñonero-Candela and Rasmussen (2005). Within this field, active subsampling strategies have been used to select information-rich data through use of information theoretic criteria (Seeger et al. 2003; Krause et al. 2008; Deisenroth et al. 2009). Our stationary work is particularly similar to Seeger et al. (2003). However, we exploit the time-sequential nature of laser data (see Section 5.3) to form an exact and inexpensive predictive distribution for use in our decision criterion.

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 (Plagemann et al. 2007). 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.

For the non-stationary extensions of our approach we chose to marginalize over unknown hyperparameters as opposed to relying on expert knowledge as in Plagemann et al. (2008). Guided by the marginal likelihood of the data we produce a general framework that is not restricted in the type of model hyperparameters.

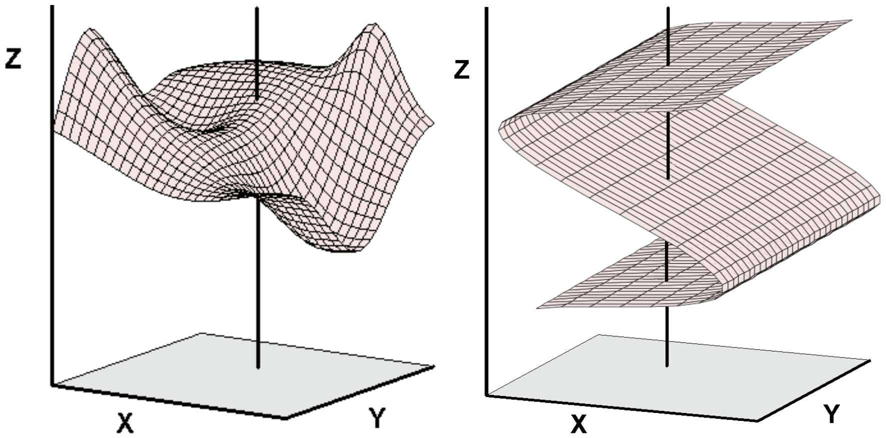


Fig. 2. We wish to map scenes with multiple elevations for a given x, y point (right), rather than being restricted to a trivial case (left).

4. Non-functional surface representation via sensor configuration space parameterization

Our aim is to form an implicit representation of workspace surfaces by processing data gathered from a laser sensor being moved through space along an arbitrary path. In particular, we pose queries of points in (x, y, z) on the workspace's surface as range queries along arbitrary rays emanating from the sensor at a point along the sensor's trajectory. Let a beam from the laser be parameterized as a point $\mathbf{q} \in \mathbb{S} \times \mathbb{R}$ where $\mathbf{q} = [\theta, t]^{\mathrm{T}}$. Here θ denotes the 1D angular position of the laser beam (*not the whole sensor unit*) and t denotes the timestamp of the laser scan.

We describe $\mathbf{q}$ as a point in *sensor configuration space* $\mathcal{Q}$, as illustrated in Figure 3, and refer to it as a particular configuration of the laser sensor. Sensor configuration space provides a natural domain for range regression since it is closely related to the state of the laser and not that of the rest of the robotic system. The mapping from sensor configuration space to range is necessarily a well-formed function: a laser sensor associates a single range r with any point in $\mathcal{Q}$ such that

$$G : \mathbb{S} \times \mathbb{R} \to \mathbb{R}, \quad \mathbf{q} \mapsto r. \tag{1}$$

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

$$E'(\mathbf{q}) = E(V(\mathbf{q}), G(\mathbf{q})) \tag{2}$$

where

$$V : \mathbb{S} \times \mathbb{R} \to \mathbb{S}^3 \times \mathbb{R}^3, \mathbf{q} \mapsto \mathbf{p} \tag{3}$$

$$E : (\mathbb{S}^3 \times \mathbb{R}^3) \times \mathbb{R} \to \mathbb{R}^3, (\mathbf{p}, r) \mapsto \mathbf{x} \tag{4}$$

For every point $\mathbf{q}$ in $\mathcal{Q}$, $V(\mathbf{q})$ provides the six-degree-of-freedom (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. Here $E(\mathbf{p}, r)$ 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. This parameterization naturally eschews the non-functional relation between elevation z and (x, y) location that is commonly encountered in terrain mapping formulations (Figure 2). By keeping each operation distinct, we also decouple robot trajectory estimation $V(\mathbf{q})$, from that of the regression of the laser data $G(\mathbf{q})$.[1] This permits independent relaxation of the sensor trajectory (which, for example, could be in response to loop closure events).

To form an estimate of $G(\mathbf{q})$ at any arbitrary position in $\mathcal{Q}$ we turn to GP regression. Given a model specified through hyperparameters ξ and a set of measurements $\mathcal{D} = \{(\mathbf{q}_i, r_i)\}_{i=1}^{N}$ we can obtain a distribution $p(r^* | \mathbf{q}^*, \xi, \mathcal{D})$ for a range prediction r^* at a query point $\mathbf{q}^*$. In the following section we present a brief summary of how this predictive distribution is obtained from a sliding window in sensor configuration space.

5. Sliding-window Gaussian process regression

GPs provide for non-parametric probabilistic regression over $\mathcal{Q}$. They allow us to marginalize over model parameters ζ that are embodied by the GP, leaving only hyperparameters ξ to specify the general shape of the regression.

$$p(r^* | \mathbf{q}^*, \xi, \mathcal{D}) = \int p(r^* | \zeta, \mathbf{q}^*, \xi, \mathcal{D}) p(\zeta | \mathbf{q}^*, \xi, \mathcal{D}) \, d\zeta. \tag{5}$$

We assume a likelihood model $p(r^* | \zeta, \mathbf{q}^*, \xi, \mathcal{D})$ with additive Gaussian noise to account for sensor inaccuracies. The GP prior $p(\zeta | \mathbf{q}^*, \xi, \mathcal{D})$ describes a distribution over latent functions underlying our observation model that we marginalize to form our posterior predictive distribution $p(r^* | \mathbf{q}^*, \xi, \mathcal{D})$. A GP is embodied by a collection of

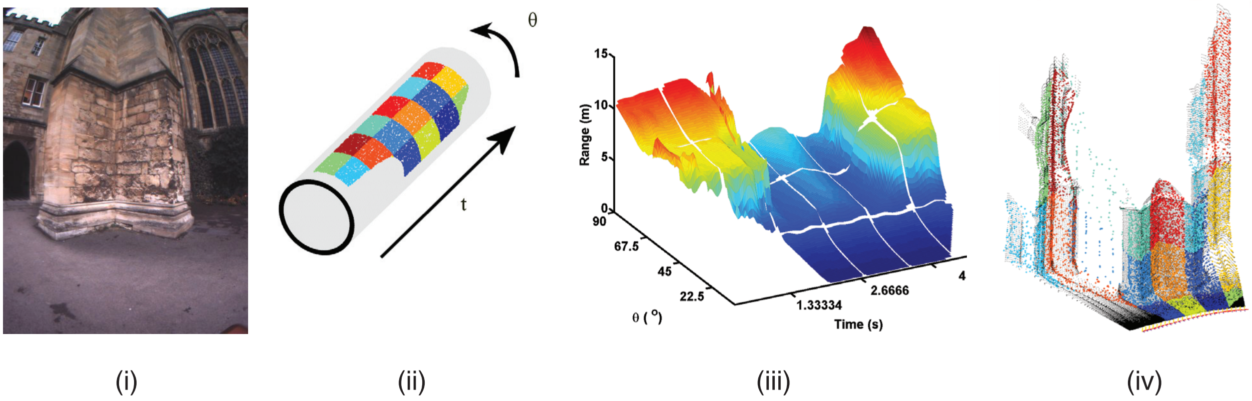


Fig. 3. Parameterization used in this work. (i) A wide-angle camera image used to provide the visual context for this example. The image is captured at the same time as the laser data plots in (ii) to (iv), which forms the central part of test case (d) in Figure 6 (see Section 8). (ii) Visualization of sensor configuration space $\mathcal{Q}$. Every point $\mathbf{q}$ on this manifold is parameterized by time t and the angular position of the laser beam θ. (iii) Laser range image. Here $\mathcal{Q}$ is unwrapped with t and θ along the horizontal axes and output of the mapping $G(\mathbf{q})$ (range measurement $\mathbf{r}$) along the vertical. (iv) 3D Euclidean space. Ranges $\mathbf{r}$ and vehicle poses $\mathbf{p}$ are mapped through $E(\mathbf{p}, \mathbf{r})$ to points in 3D. The robot trajectory is shown as coloured frames across the bottom. A coloured tiling is used to the aid understanding of the mapping between sensor configuration space and Euclidean space.

jointly Gaussian distributed random variables that is fully specified by mean $\mu(\mathbf{q})$ and covariance $k(\mathbf{q}, \mathbf{q}')$ functions. In our application we are concerned with estimating the mapping $G(\mathbf{q})$ that corresponds to these latent functions. Given known training observations $\mathbf{r} = \{r_i\}_{i=1}^N$ and $\mathcal{Q} = \{\mathbf{q}_i\}_{i=1}^N$, and known $\mathbf{q}^*$ with corresponding unknown target r^*, we can write

$$\begin{bmatrix}\mathbf{r}\\ r^*\end{bmatrix} \sim \mathcal{N}\left(\begin{matrix}\mu(\mathcal{Q})\\ \mu(\mathbf{q}^*)\end{matrix}, \begin{bmatrix} K_{\mathbf{r}} & \mathbf{k}(\mathcal{Q},\mathbf{q}^*)\\ \mathbf{k}(\mathcal{Q},\mathbf{q}^*)^{\mathrm{T}} & k(\mathbf{q}^*,\mathbf{q}^*)+\sigma_m^2\end{bmatrix}\right), \qquad (6)$$

where

$$K_{\mathbf{r}} = K(\mathcal{Q}, \mathcal{Q}) + \sigma_m^2 I. \qquad (7)$$

Each element of K is, in the stationary case, a suitably chosen distance metric $d = \|\mathbf{q} - \mathbf{q}'\|$ between two corresponding points $\mathbf{q}$ and $\mathbf{q}'$ in $\mathcal{Q}$. In Equations (6) and (7) we explicitly account for the measurement noise of our likelihood model through an additive white noise process of strength σ_m along the diagonal entries of K (see Section 8.5 for an investigation into the effect of σ_m). The derivation of the mean $\mathbb{E}[r^*]$ and covariance $\mathbb{V}[r^*]$ of $p(r^*|\mathbf{q}^*, \xi, \mathcal{D})$ (Equation (5)) for a deterministic $\mu(\mathbf{q}) = 0$ (as is commonly used Vasudevan et al. (2009)) are standard and can be found, for example, in Rasmussen and Williams (2006):

$$\mathbb{E}[r^*] = \mathbf{k}(\mathcal{Q}, \mathbf{q}^*)^{\mathrm{T}} K_{\mathbf{r}}^{-1} \mathbf{r} \qquad (8)$$

$$\begin{aligned}\mathbb{V}[r^*] = {} & k(\mathbf{q}^*, \mathbf{q}^*) + \sigma_m^2 - \\ & \mathbf{k}(\mathcal{Q}, \mathbf{q}^*)^{\mathrm{T}} K_{\mathbf{r}}^{-1} \mathbf{k}(\mathcal{Q}, \mathbf{q}^*) .\end{aligned} \qquad (9)$$

Throughout this work we use a member of the Matérn class of covariance functions as advocated in Stein (1999). We note, however, that many others, including non-stationary covariance functions (Lang et al. 2007), could be adapted and substituted in its place. The Matérn class of functions is dependent on a shape parameter ν which regulates the smoothness of the interpolation. It equates to the more standard exponential covariance function as a special case when $\nu = \frac{1}{2}$, and the squared exponential as $\nu \to \infty$. As suggested by Rasmussen and Williams (2006), we explored several common choices of $\nu = \left\{\frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \infty\right\}$ over a number of workspaces varying in complexity. We found that $\nu = \frac{3}{2}$ consistently produced accurate surface reconstructions. Although the smoother covariance functions $\nu = \left\{\frac{5}{2}, \infty\right\}$ performed well for simple workspaces, they seemed over constrained in complex scenarios, and vice-versa for the rough covariance function $\nu = \frac{1}{2}$. Thus

$$k(\mathbf{q}, \mathbf{q}')_{\text{matern } \nu=\frac{3}{2}} = \sigma_p^2 (1 + \frac{d\sqrt{3}}{l}) \exp\left(-\frac{d\sqrt{3}}{l}\right), \qquad (10)$$

where σ_p^2 is the process noise and $d = \|\mathbf{q} - \mathbf{q}'\|_{\mathcal{Q}}$ denotes the geodesic distance between $\mathbf{q}$ and $\mathbf{q}'$ in $\mathcal{Q}$. For this sensor configuration d is the ℓ_2 norm, and we use a single length scale l to capture the covariance of our regression domain.[2] The result is an isotropic covariance that we use in the stationary version of this algorithm (see Section 8).

5.1. Sliding window approach

The quantities of data we consider render the application of a single monolithic GP infeasible: time complexity of a naïve implementation of GP regression is cubic in N, the size of the dataset $\mathcal{D}$. Instead, for each prediction we enforce a predictive support window $\widetilde{\mathcal{D}}$ of *fixed* size n, formed from the closest (in terms of the ℓ_1 norm from t to t^*)

laser measurements that have been actively accepted by our algorithm. This guarantees a constant time algorithm. The time-sequential nature of laser data ensures that this support window slides across $\mathcal{Q}$ as the robot progresses along its trajectory. The span η of $\widetilde{\mathcal{D}}$ is defined as the index of the newest measurement to have been considered, minus the index of the oldest measurement in $\widetilde{\mathcal{D}}$. Although n is fixed, our algorithm will allow η to vary significantly in accordance with workspace complexity (see Figure 17).

5.2. Stochastic mean function

We affect a stochastic non-zero mean function $\mu(\mathbf{q})$ through the use of explicit basis functions (Rasmussen and Williams 2006), $\mathbf{h}(\mathbf{q})$ with weightings $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{b}, B)$. Common examples include polynomials $\{1, x, x^2, \dots\}$ and sinusoids $\{1, \sin(x), \cos(x), \sin(2x), \dots\}$. These are superimposed with the zero-mean GP $Z(\mathbf{q}) \sim \mathcal{GP}(0, k(\mathbf{q}, \mathbf{q}'))$ of Section 5

$$G(\mathbf{q}) = Z(\mathbf{q}) + \mathbf{h}(\mathbf{q})^{\mathrm{T}} \boldsymbol{\beta}. \tag{11}$$

In the absence of data, predictions using $G(\mathbf{q}) = p(r^*|\mathbf{q}^*, \xi, \widetilde{\mathcal{D}})$ tend to the surface estimates of the explicit basis functions, rather than that of the zero mean GP. One may view this formulation as the GP modelling the residuals between the implicit surface generated by $\mathbf{h}(\mathbf{q})^{\mathrm{T}} \boldsymbol{\beta}$, and $\widetilde{\mathcal{D}}$. It has been used successfully in robotics applications such as Nguyen-Tuong and Peters (2010) and is particularly beneficial when predicting near the boundary of our active window. It is the key to our compression rate (Section 5.3).

We take the limiting case as the prior on the weightings tends towards a uniform distribution ($B^{-1} \rightarrow \mathbf{0}$). Following O'Hagan and Kingman (1978) we arrive at a predictive distribution that is independent of $\mathbf{b}$:

$$\mu_G(\mathbf{q}^*) = \mathbb{E}_G[r^*] = \mathbb{E}_Z[r^*] + \mathbf{u}^{\mathrm{T}} \bar{\boldsymbol{\beta}}, \tag{12}$$

$$\Sigma_G(\mathbf{q}^*) = \mathbb{V}_G[r^*] = \mathbb{V}_Z[r^*] + \mathbf{u}^{\mathrm{T}} A^{-1} \mathbf{u}, \tag{13}$$

where $\mathbb{E}_Z[r^*]$ and $\mathbb{V}_Z[r^*]$ are given by Equations (8) and (9), respectively, and

$$\bar{\boldsymbol{\beta}} = A^{-1} H K_{\mathbf{r}}^{-1} \mathbf{r}, \tag{14}$$

$$\mathbf{u} = \mathbf{h}(\mathbf{q}^*) - H K_{\mathbf{r}}^{-1} \mathbf{k}(Q, \mathbf{q}^*), \tag{15}$$

$$A = H K_{\mathbf{r}}^{-1} H^{\mathrm{T}}, \tag{16}$$

$$H = [\mathbf{h}(\mathbf{q}_1), \mathbf{h}(\mathbf{q}_2), \dots, \mathbf{h}(\mathbf{q}_n)]. \tag{17}$$

This formulation affords us the benefit of explicit basis functions without requiring additional parameters, and maintains the desirable predictive properties of GPs. Like Equations (8) and (9) an inverse must be calculated, A^{-1}, but its dimension is that of the number of explicit basis functions chosen for the application, and not the support set size, n. We choose the polynomial set of basis functions $\mathbf{h} = [1, \theta, \theta^2, \theta^3, t, t^2, \theta t, \theta t^2, \theta^2 t, \theta^2 t^2]^{\mathrm{T}}$ and have found cross terms above second order to have marginal overall impact on predictions.

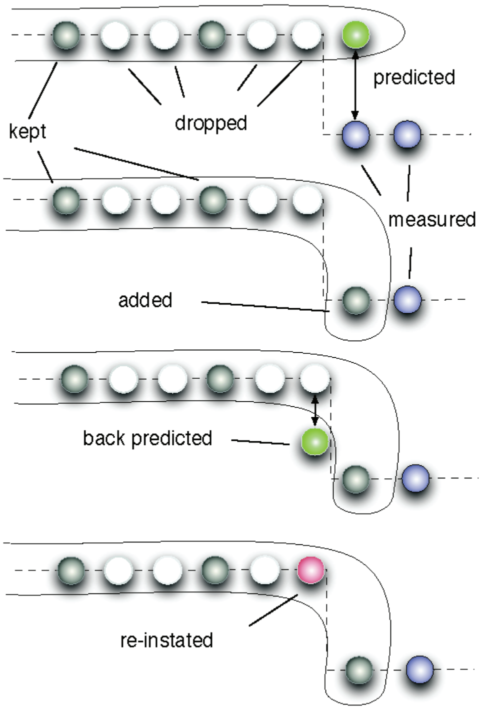


Fig. 4. Active sampling of a 1D data series. Each node, connected by the chained lines, represents a measurement. Black indicates data accepted by the forward pass of the active sampling algorithm while the white have been rejected and the blue are yet to be observed. Green nodes are predictions made 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 that has a KL divergence from the measurement 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.

5.3. Active sampling through KL divergence

In this section we describe our method for adaptive laser measurement selection. We decide if our prediction of a single prediction $p(r^*|\mathbf{q}^*, \widetilde{\mathcal{D}}) = \mathcal{N}(\mu_G, \sigma_G^2)$ (Equations (12) and (13)) differs significantly from a measurement $r_m \sim \mathcal{N}(\mu_m, \sigma_m^2)$ at $\mathbf{q}^*$. We use the KL divergence between the two distributions to ascertain whether 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 $|\mathcal{D}|/|\mathcal{D}'|$ where $\mathcal{D}'$ is the actively selected subset of the dataset $\mathcal{D}$. Thorough analysis on the choice of all model parameters can be found in Sections 8 and 10. The KL divergence between two 1D Gaussian distributions has a closed form solution:

$$D_{KL}(\mathcal{N}_m||\mathcal{N}_G) = \frac{1}{2}\left[\log_e \frac{\sigma_G^2}{\sigma_m^2} + \frac{\mu_G^2 + \mu_m^2 - 2\mu_G\mu_m + \sigma_m^2}{\sigma_G^2} - 1\right]. \tag{18}$$

On adoption of any new laser measurement into the active support set we introspectively analyse the closest (ℓ_1 norm in $\mathcal{Q}$) r_m that had previously been deemed redundant to determine whether 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 4. Introspection allows both sides of discontinuities to be analysed given the final sampling of the opposite side of the discontinuity. Figure 5 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 section, 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 (Rusinkiewicz and Levoy 2001).

6. Implementation considerations

We seek to maximize the support set size for a given computational capacity. This provides more fully conditioned predictions, over a greater prediction window size, both while $\mathcal{D}'$ is constructed online and for subsequent surface querying. To this end we use Cholesky decomposition and rank one Cholesky updates to implicitly perform the expensive inversion of $K_{\mathbf{r}} = C_{\mathbf{r}}^{\mathrm{T}} C_{\mathbf{r}}$ (Equation (7)), as is common in the GP literature (Scholkopf and Smola 2002; Osborne et al. 2008; Seeger 2008). We also exploit the upper triangular structure of $C_{\mathbf{r}}$ where possible and ensure that recurring calculations are performed once. For example, we update $C_{\mathbf{r}}^{-\mathrm{T}}\mathbf{r}$ in Equations (8), (14), (35), (36) and (37) by using backwards substitution rather than recalculate it from scratch. We also make the following considerations to the practical construction of our algorithm.

6.1. Polynomial basis function conditioning

Although we use the Cholesky decomposition of A (Equation (16)) and use only low-order polynomial basis functions, the conditioning of A can become prohibitive when this algorithm is applied to very large datasets. Fortunately, the only non-stationary element of our stochastic GP (Section 5.2) is the set of polynomial basis functions $\mathbf{h}$. This means that we can shift $\mathcal{Q}$ for the construction of H to reduce the conditioning of A whilst leaving the rest of our algorithm unchanged. Specifically we use H' to replace H in all equations:

$$H'(\mathcal{Q}) = H(\mathcal{Q} - \bar{\mathcal{Q}}), \tag{19}$$

where

$$\bar{\mathcal{Q}} = [\bar{\theta}, \bar{t}]^{\mathrm{T}}. \tag{20}$$

Here $\bar{\theta}$ and $\bar{t}$ are the means of the current active set in each dimension of $\mathcal{Q}$. When computational savings are required, $\bar{\mathcal{Q}}$ can be updated only when the conditioning of A becomes poor. Between updates of $\bar{\mathcal{Q}}$, we recycle elements of H as long as the corresponding measurements remain within $\widetilde{\mathcal{D}}$. The worst performing ξ_i (see Section 9) required the update of $\bar{\mathcal{Q}}$ once in each of our test cases for the results presented in Section 10.

6.2. Marginalization and expansion

Each time we accept a new measurement into our active set we first marginalize the most stale measurement (at the back of the active window), then expand the active set to include the new measurement. We perform both operations as an atomic unit allowing us to perform them in place, rather than expanding and shrinking the matrix $C_{\mathbf{r}}$. Overall this reduces the dominant complexity of our algorithm from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$, where n is the size of our active window $\widetilde{\mathcal{D}}$.

6.2.1. Marginalization Given a positive definite matrix K, and its upper triangular Cholesky factor C such that $C^{\mathrm{T}}C = K$

$$K = \begin{bmatrix} K_{1,1} & \mathbf{k}_{1,2} & K_{1,3} \\ \mathbf{k}_{1,2}^{\mathrm{T}} & k_{2,2} & \mathbf{k}_{2,3} \\ K_{1,3}^{\mathrm{T}} & \mathbf{k}_{2,3}^{\mathrm{T}} & K_{3,3} \end{bmatrix}, \quad C = \begin{bmatrix} C_{1,1} & \mathbf{c}_{1,2} & C_{1,3} \\ \mathbf{0} & c_{2,2} & \mathbf{c}_{2,3} \\ \mathbf{0} & \mathbf{0} & C_{3,3} \end{bmatrix}$$

we can remove the central row and column by marginalizing C to

$$\begin{bmatrix} C_{1,1} & C_{1,3} \\ \mathbf{0} & \Upsilon(C_{3,3}^{\mathrm{T}} C_{3,3} + \mathbf{c}_{2,3}^{\mathrm{T}} \mathbf{c}_{2,3}) \end{bmatrix}, \tag{21}$$

where Υ is the Cholesky update of Dongara et al. (1979) which exploits the special structure of $\mathbf{c}_{2,3}^{\mathrm{T}}\mathbf{c}_{2,3}$, and is of complexity $O(n^2)$. An efficient implementation can be found in, for example, *cholupdate* of MATLAB®.

6.2.2. Expansion Given positive definite K and its corresponding upper triangular Cholesky factor R,

$$K = \begin{bmatrix} K_{1,1} & K_{1,3} \\ K_{1,3}^{\mathrm{T}} & K_{3,3} \end{bmatrix} R = \begin{bmatrix} R_{1,1} & R_{1,3} \\ \mathbf{0} & R_{3,3} \end{bmatrix}$$

we wish to expand to C (above) through insertion of a central row and column:

$$\begin{bmatrix} R_{1,1} & R_{1,1}^{\mathrm{T}} \setminus \mathbf{k}_{1,2} & R_{1,3} \\ \mathbf{0} & \sqrt{k_{2,2} - \mathbf{c}_{1,2}^{\mathrm{T}}\mathbf{c}_{1,2}} & \frac{\mathbf{k}_{2,3} - \mathbf{c}_{1,2}^{\mathrm{T}} C_{1,3}}{c_{2,2}} \\ \mathbf{0} & \mathbf{0} & \Upsilon(R_{3,3}^{\mathrm{T}} R_{3,3} - \mathbf{c}_{2,3}^{\mathrm{T}}\mathbf{c}_{2,3}) \end{bmatrix}, \tag{22}$$

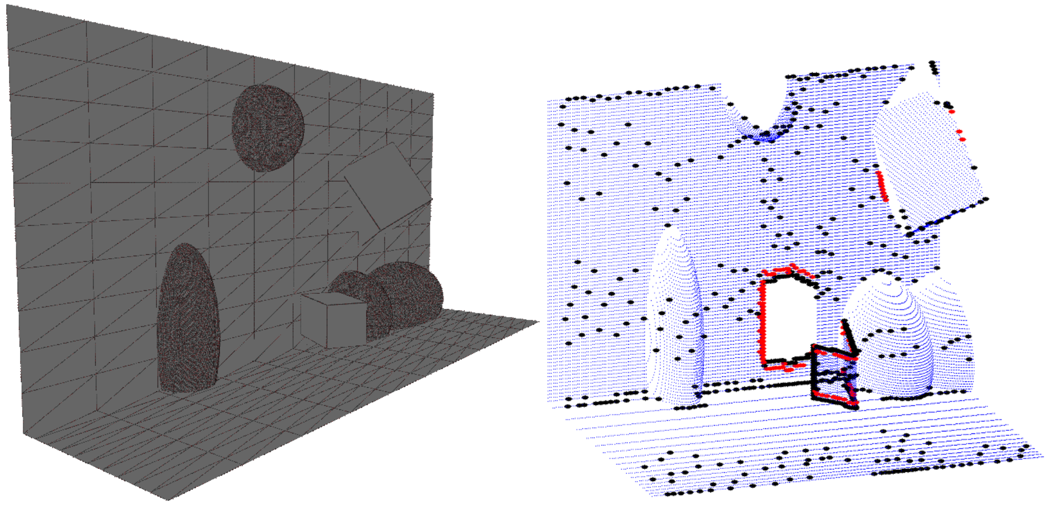


Fig. 5. Result from using active sampling within our sensor configuration space formulation. (Left) CAD model used to generate synthetic data based on the wall section (7 m × 3 m × 1.5 m) of test case (a) (see Figure 6) with added objects. (Right) Plot of synthetic data as a laser scanner is dragged 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 and 11). 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 8.

where we have used \ to indicate where backwards substitution is used in the solution of $\mathbf{c}_{1,2}$ for the equation $R_{1,1}^{\mathrm{T}}\mathbf{c}_{1,2} = \mathbf{k}_{1,2}$.

6.3. *Forming predictions*

In addition to predictions that are made as part of active sampling at the front of $\widetilde{\mathcal{D}}$ we can form predictions at arbitrary locations within $\mathcal{Q}$. In Sections 8 and 10 we present results that use predictions formed by $\widetilde{\mathcal{D}}$ as the subset is established. In this online formulation each prediction $r^* \sim \mathcal{N}(\mu_G, \sigma_G^2)$ is made as the mean time $\bar{t}$ of the active window Q passes the corresponding t^* in $\mathcal{Q}$. This ensures that predictions are made away from the front edge of the active subset, where the predictive surface undergoes frequent modification as part of the sampling process. Predictions are made after each round of introspection so that we capture both sides of any discontinuities as explained in Section 5.3. On occasions when the span of the active window η collapses (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 ensures predictions remain within the extent of the active subset. In particular, we check to see whether the αth measurement from the back of the active set is older than the last t^*, to ensure that all predictions are made with sufficient support.

For offline prediction of r^* the active subset surrounding $\mathbf{q}^*$ can be constructed from the closest measurements (in terms of the ℓ_1 norm from t to t^*), as noted in Section 5.1. Offline predictions are guaranteed to be made after all introspections, and the active patch can be reused for any further predictions r^* in the vicinity.

7. Experimental setup

Our aim is to analyse how changing the KL divergence threshold κ affects the compression rate of point clouds collected in a commonplace urban environment. Real-world examples (Figure 6) are from the New College dataset (Smith et al. 2009). Data were gathered from a two-wheeled Segway RMP~200 platform with two vertically aligned SICK LMS291-S14 laser sensors mounted on the side of the vehicle. Although the vehicle trajectory is not used in our algorithm, it allows us to project our results into Euclidean space for plotting, as per Equation (4). Pose estimates were obtained using visual odometry from a forward-facing stereoscopic camera (Newman et al. 2009). We have also used synthetic data, which we have generated from two CAD models and a sensor model of a laser scanner. Test case (e) is a simplification of test case (a) in Figure 6, where we represent the wall and floor by vertical and horizontal planes of the same proportions, and approximate the laser scanners trajectory as a single straight line. A more complex model is used for test case (f) (Figure 5), based on that used for test case (e) but with added objects in the scene.

Our success metrics are the reconstruction error (in metres) and compression rate $|\mathcal{D}'|/|\mathcal{D}|\,\%$. Error is defined as the ℓ_1 norm in range between a hold out set of laser measurements and predictions, made online as $\mathcal{D}'$ is established (as per Section 6.3). Compression rate is defined as the percentage of the laser measurements that are used

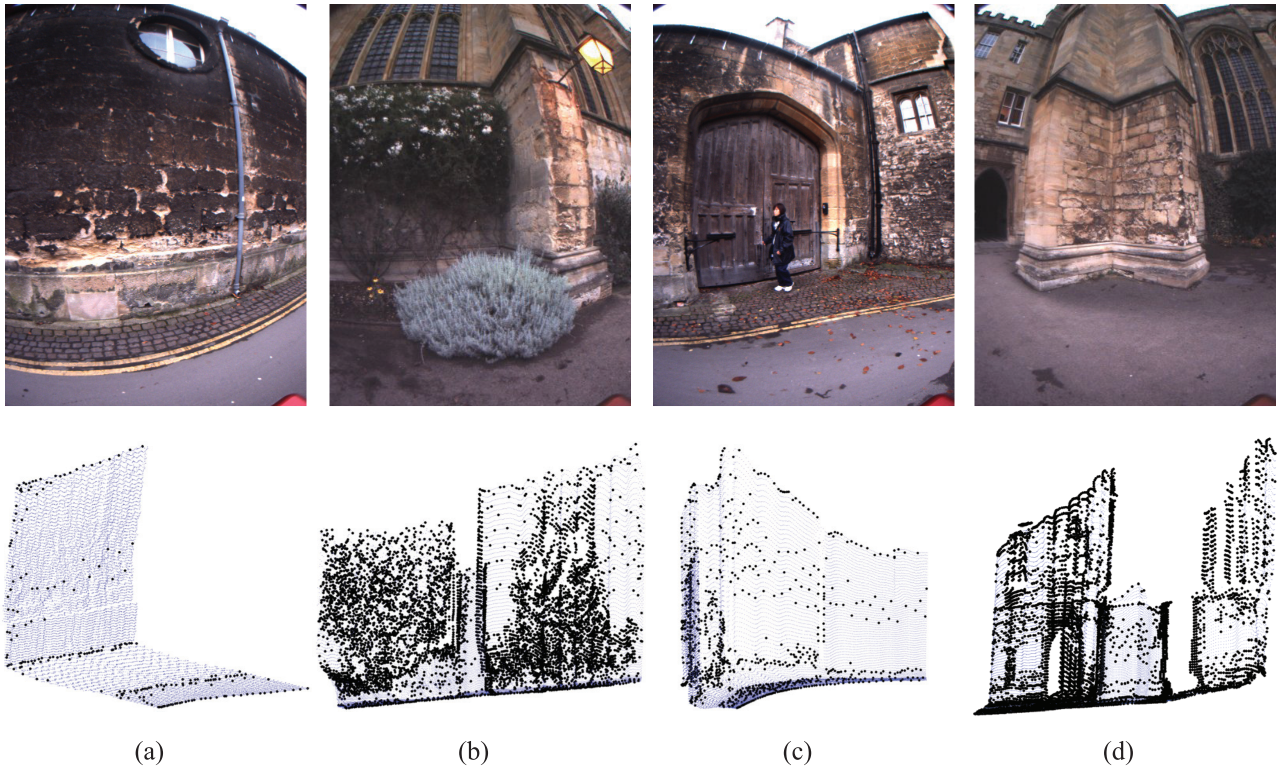


Fig. 6. Real-world test cases (column-wise). (Top) Images were captured at the same time as the laser data in blue (bottom). Note that the field of view of each sensor does not match and there is considerable distortion in the images due to the use of a wide-angle lens. This is most evident in test case (a) where the window in the image is not present in the laser data. (Bottom) Blue measurements indicate the original laser measurements $\mathcal{D}$ and black the actively sampled measurement $\mathcal{D}'$ in Euclidean space for $\kappa = 0.8\,nats$ as detailed in Section 8. One can clearly observe that intersections of planes, a person, bushes and other complicated regions of $\mathcal{W}$ are more heavily sampled than simple regions.

in our representation over the total number of laser measurements. Therefore, a high decimation factor $|\mathcal{D}|/|\mathcal{D}'|$ (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.

We have used an active window size $n = 200$ and $\alpha = 5$ in all of our experiments. These parameters were found empirically to produce consistently accurate surface representations for a given compression rate, as discussed in Sections 5 and 6.3.

8. Stationary Gaussian process results

In addition to the model parameters discussed in Section 7 we use the Matérn class of covariance function and choose measurement variance $\sigma_m^2 = 0.01m^2$ and hyperparameter values $\xi = \{l, \sigma_p^2\}$ with length scale $l = 8$ units and process variance $\sigma_p^2 = 0.05$ m^2. We first analyse how compression rate and reconstruction error (Sections 8.2 and 8.1) then computation time (Section 8.3) vary with our prediction threshold κ. For completeness we also include a sensitivity analysis of our choice of hyperparameter values in Section 8.4, and in Section 8.5 explore the effect of varying the likelihood model.

8.1. Compression rate

Figure 7 depicts typical subsampling results as κ is varied. For aggressive compression rate $|\mathcal{D}'|/|\mathcal{D}|\,\%$ 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.

In Figure 8 results are collated across a range of κ for the six test cases. Intuitively, as the threshold is increased, lower compression rates $|\mathcal{D}'|/|\mathcal{D}|\,\%$ are produced. Lower compression rates are accompanied by an increase in error as depicted by the box plots in Figure 9. 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 with that of the 1 m discontinuity that the person in (c) presents. In all cases, these complex regions have been sampled most heavily: the

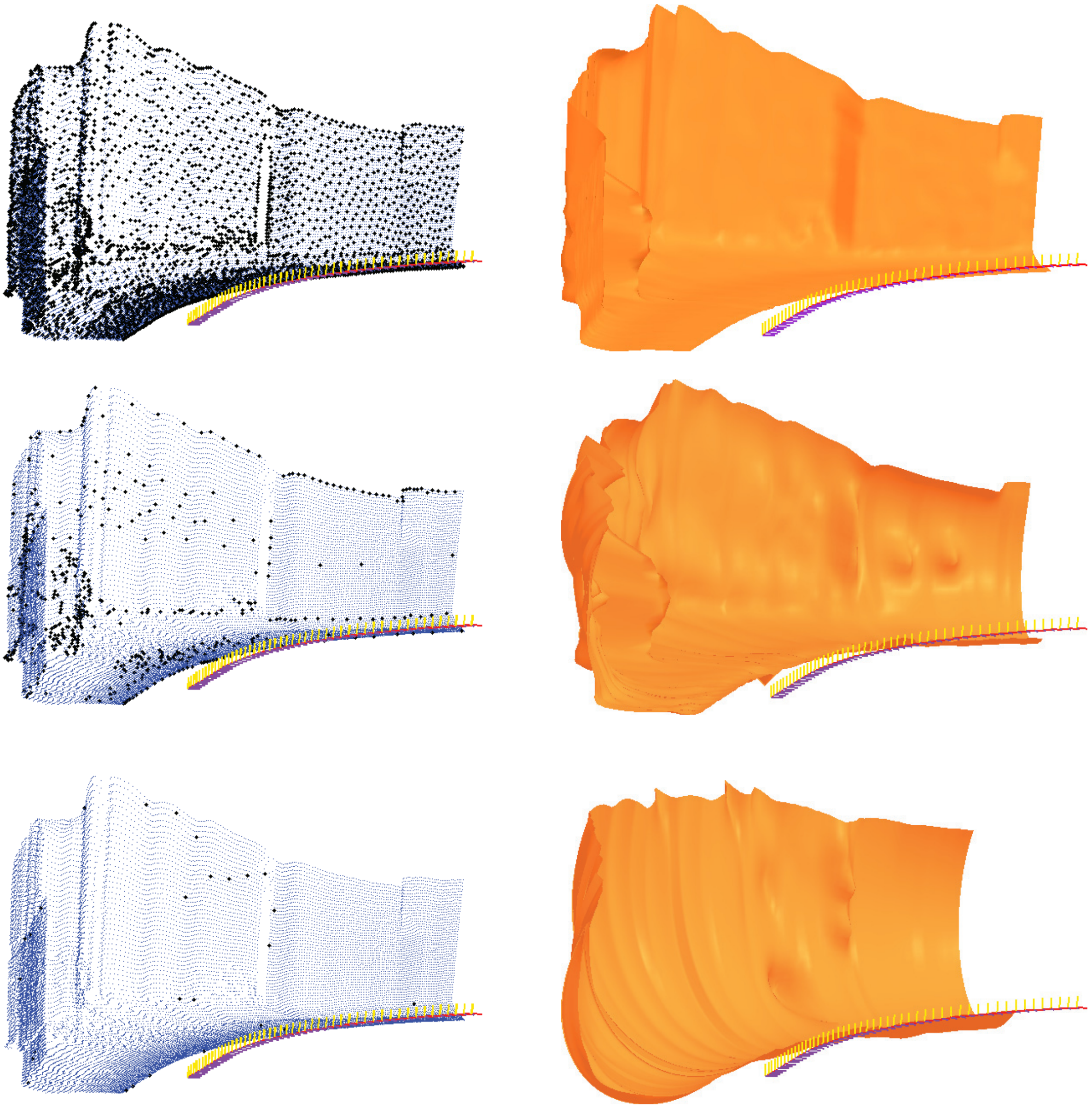


Fig. 7. Laser measurement subsample plots for various values of our KL divergence threshold κ for test case (c) of Figure 6, plotted in Euclidean x, y, z space. Each row corresponds to $\kappa = \{0.3, 1.26, 3.18\}$ *nats* working downwards. (Left) The original 28,892 measurements are in blue and measurements 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 Section 6.3. 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) 4,543 laser measurements are used to produce predictions resulting in a mean error of 0.014 m while; (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.

outline of the person can be recognized in the Euclidean plot of (c) in Figure 6.

8.2. Surface error

Box plots indicate surface error distributions for each of the test cases in Figure 9. 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 10.

To demonstrate the spatial distribution of error we provide the surface predictions of test cases (a) and (b) coloured according to error in Figure 11. For test case (b)

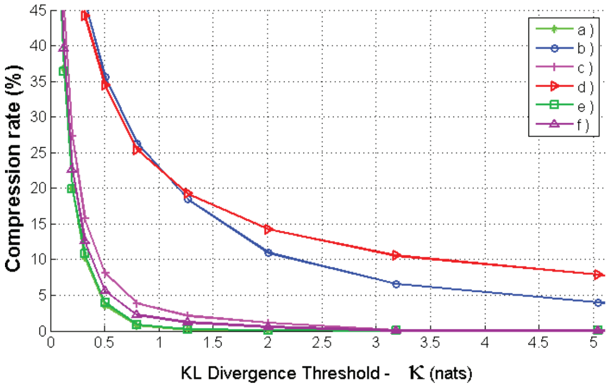


Fig. 8. Compression rate versus κ for a variety of urban and synthetic environments. Compression rate, measured as the percentage of $\mathcal{D}$ selected by our algorithm ($|\mathcal{D}'|/|\mathcal{D}|\,\%$), is plotted against κ at each of the configurations denoted by markers. A lower compression rate results from higher thresholds and scenes that are less complex. See Section 7 for a discussion of the test cases.

we observe that the worst errors occur at discontinuities and for very noisy surfaces 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 5.3.

8.3. *Timing considerations*

We also analysed the effect of varying KL divergence threshold κ on the time taken to compute our surface representation. Each of our examples (Figure 6) cover roughly 9 seconds (120,000 measurements) of laser data of which we uniformly sample to use a quarter as input to our algorithm. In Figure 12 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 unoptimized MATLAB® code on 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 adopting all measurements into the active window, and 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.

8.4. *Choosing hyperparameters*

Our hyperparameters ξ consist of length scale l and process variance σ_p^2. To chose values for these 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. The cost function used for our optimization aimed to minimize both mean error and compression rate. 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 $\sigma_p^2 = 0.05$ m^2 as they provided consistently accurate surface representations with aggressive compression rates.

8.5. *Varying the likelihood model*

We now turn our attention to the dependence of our results to our likelihood model $p(r^*|\zeta, \mathbf{q}^*, \xi, \mathcal{D})$ (see Equation (5)) which is constructed using measurement variance σ_m^2. For a given κ one would expect that, as σ_m^2 decreases, the compression rate $|\mathcal{D}'|/|\mathcal{D}|\,\%$ would increase. The KL divergence (Equation (18)) between two distributions with given means is greater for smaller variances. Hence more measurements are adopted into the active set which is demonstrated in Figure 13 with the migration of the curves to the right as σ_m^2 decreases. We include Figure 14 with $\sigma_m^2 = 1 \times 10^{-4}$ m^2 rather than $\sigma_m^2 = 1 \times 10^{-2}$ m^2 that is used in previous sections, to allow comparison of the resulting errors.

9. Non-stationarity though a Gaussian mixture model

Thus far our approach has considered the use of a stationary GP. While we have shown that this approach can produce aggressive compression rates for a relatively small loss in accuracy, we would like to increase the surface prediction accuracy for an even lower set of retained measurements $\mathcal{D}'$. We achieve this by allowing our algorithm to be *non-stationary* by automatically selecting hyperparameter ξ values in accordance with the local complexity of the workspace. To illustrate the importance of this consider increasing the length scale l in a region of sensor configuration space $\mathcal{Q}$. In this region a laser measurement will influence predictions from a greater distance in $\mathcal{Q}$. This creates a smoother predictive surface which is more

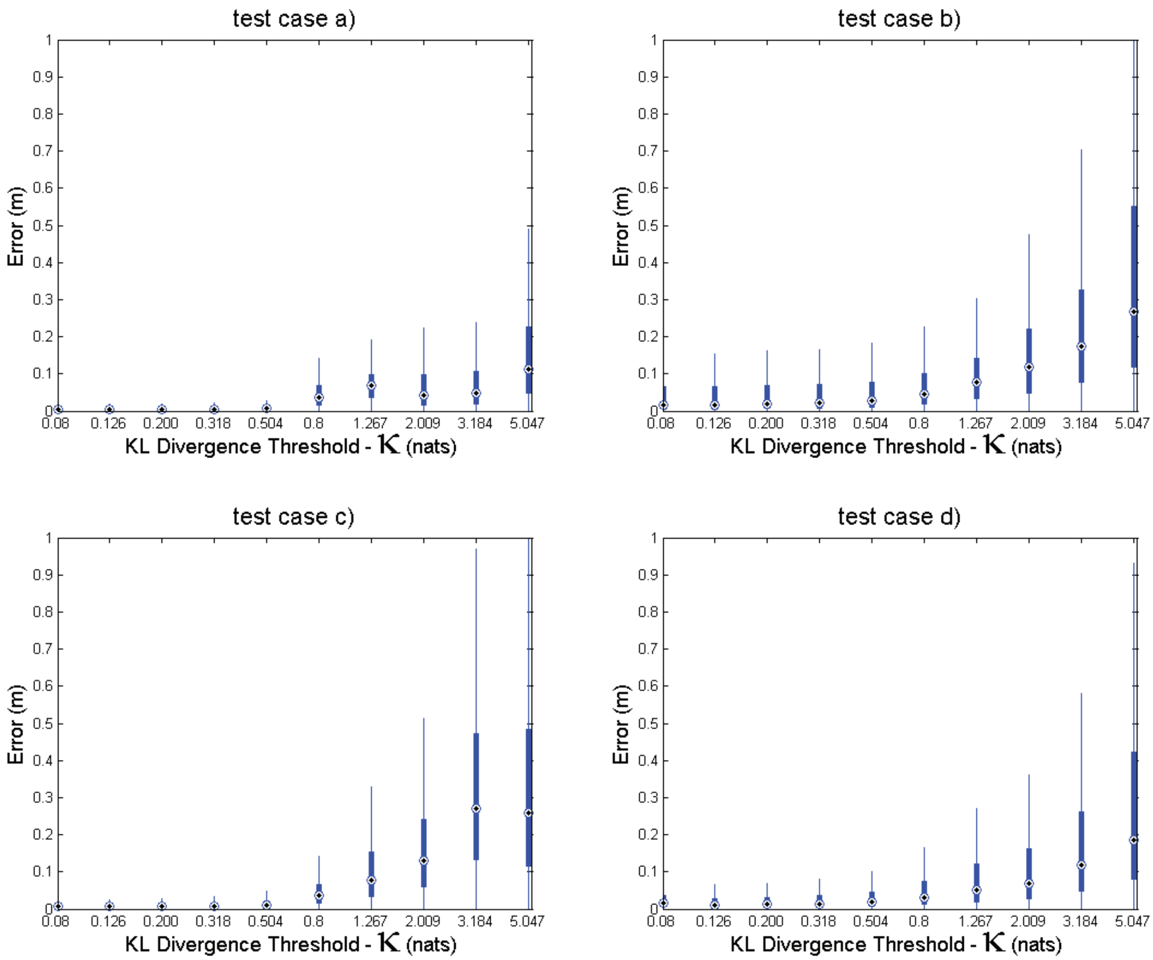


Fig. 9. Error versus prediction threshold κ for test cases (a)–(d) as detailed in Section 7. Box plots denote median and interquartile range for each of the logarithmically spaced κ indicated by markers in Figure 8. Whiskers are plotted that extend a maximum of one and a half times the interquartile range. 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. Scenes with a high complexity, such as test case (b) start with a significantly higher median error 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.

applicable in regions of low complexity than those surfaces generated from shorter length scales. As each measurement has more effect on predictions, fewer measurements are required per square area of $\mathcal{Q}$ to support predictions and the span of the active patch η increases. We can therefore use fewer measurements in our representation of the region.

To capture the fact that the workspace complexity in $\mathcal{Q}$ is generally non-isotropic we choose to update the covariance function $k(\mathbf{q}, \mathbf{q}')$ of Section 5. We then introduce the full non-stationary extension, which is based on a Bayesian quadrature approach. We suggest two alternative approximations that remove the prohibitive computational cost of the full approach whilst retaining a coarse degree of automatic control of the hyperparameters. Finally, the additional quadrature distribution $p(\xi)$ is discussed and implementation detail is provided.

9.1. Modifications to $k(\mathbf{q}, \mathbf{q}')$

We create our anisotropic Matérn $k(\mathbf{q}, \mathbf{q}')$ with hyperparameters $\xi = \{\sigma_p^2, l_t, l_\theta\}$ through the modification of our distance metric d in Equation (10). Our distance metric becomes that of a Mahalanobis distance through the inclusion of the length scales $\{l_\theta, l_t\}$ (as commonly found in the GP literature (Rasmussen and Williams 2006)):

$$d = \|\mathbf{q} - \mathbf{q}'\|_{\mathcal{Q}} = \sqrt{(\mathbf{q} - \mathbf{q}')^{\mathrm{T}} \begin{bmatrix} l_\theta^2 & \mathbf{0} \\ \mathbf{0} & l_t^2 \end{bmatrix}^{-1} (\mathbf{q} - \mathbf{q}')}$$

$$k(\mathbf{q}, \mathbf{q}')_{\text{matern } \nu = \frac{3}{2}} = \sigma_p^2 (1 + d\sqrt{3}) \exp(-d\sqrt{3}) . \qquad (23)$$

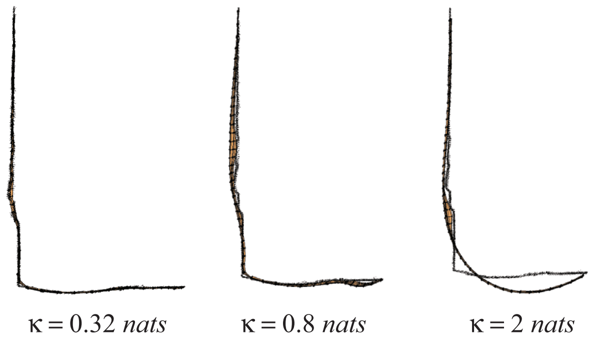


Fig. 10. Profile view of a short section of wall and ground plane from test case (a) (Figures 6 and 11), with $\kappa = \{0.32, 0.8, 2\}$ *nats* from left to right. Here $\mathcal{D}$ is shown in black, while predictions conditioned on $\widetilde{\mathcal{D}}$ are collated into the surfaces. Larger thresholds create smaller compression rates, hence smoother surfaces that are in greater error.

We use this simple augmentation to allow comparison with the results obtained using the stationary approach (Section 8). However, unlike other approaches such as that of Plagemann et al. (2008), there are no restrictions on the type of hyperparameters that can be used in this framework, so long as the covariance function is stationary. For example, we might boost the regression performance for this dataset (Smith et al. 2009) through the addition of a sinusoidal covariance term (with amplitude and period included in ξ). This term would capture the cyclical artifact in some sections of $\mathcal{Q}$ that is due to the rocking motion of the balancing platform used for data capture.

9.2. Bayesian approach to non-stationarity

We aim to achieve the automatic control of hyperparameters ξ. Rather than dealing with ξ directly we marginalize this unknown variable to form our predictive distribution $p(r^*|\mathbf{q}^*, \widetilde{\mathcal{D}})$ in the same way as we did with parameters ζ in Equation (5). To achieve this model averaging we rearrange the joint probability of the predictions r^*, data $\widetilde{\mathcal{D}}$ and hyperparameters ξ, and marginalize over ξ to arrive at

$$p(r^*|\mathbf{q}^*, \widetilde{\mathcal{D}}) = \frac{\int p(r^*|\mathbf{q}^*, \xi, \widetilde{\mathcal{D}})\, p(\widetilde{\mathcal{D}}|\xi)\, p(\xi)\; d\xi}{\int p(\widetilde{\mathcal{D}}|\xi)\, p(\xi)\; d\xi}, \tag{24}$$

where $p(r^*|\mathbf{q}^*, \xi, \widetilde{\mathcal{D}})$ is our stationary GP prediction given in Equation (5). The marginal likelihood of the data given hyperparameters ξ, $p(\widetilde{\mathcal{D}}|\xi)$ can be readily calculated (see the Appendix). Like all Bayesian techniques, we explicitly account for workspace knowledge in the prior $p(\xi)$. Unlike in Equation (5), we do not have an analytic solution to either of the marginalizations in Equation (24). Instead we turn to quadrature techniques and approximate the integrals by drawing *hyperparameter samples* ξ_i from each of the distributions over hyperparameters $p(r^*|\mathbf{q}^*, \xi, \widetilde{\mathcal{D}})$ and $p(\widetilde{\mathcal{D}}|\xi)$.

To approximate our integral we extend our Bayesian approach through the use of Bayesian Monte Carlo (O'Hagan 1991; Rasmussen and Ghahramani 2003; Osborne 2010). This technique provides a sampling method that, unlike classical Monte Carlo (MacKay 2003), accounts for both the sample values and their position through the use of uncertainty as advocated in O'Hagan (1987). Uncertainty arises in the integration since the integrand is expensive and cannot be evaluated at every location. Therefore, a prior is placed over the integrand and the unknown quantity of interest (in this case the posterior predictive distribution $p(r^*|\mathbf{q}^*, \widetilde{\mathcal{D}})$ in Equation (24)) is treated as a random variable. The prior is combined with observations of the integrand to produce a posterior distribution. A GP is used to provide a convenient way of putting a prior over functions as we have seen in Equation (5). For Equation (24) this requires placing two GP priors X and Y over $p(r^*|\mathbf{q}^*, \xi, \widetilde{\mathcal{D}})$ and $p(\widetilde{\mathcal{D}}|\xi)$, respectively.

Osborne (2010) notes the correlations induced by $p(\widetilde{\mathcal{D}}|\xi)$ appearing in both the numerator and the denominator in Equation (24) imply that Bayesian quadrature must be performed on the equation as a whole and he called the resulting technique *predictive Bayesian quadrature*. Gaussian priors were mainly used although discussion and derivations for other priors were also provided. The sequential form of the algorithm additionally exploits a maximum *a posteriori* approximation of Y to permit a computationally tractable solution. The overall predictive distribution is affected by each GP regressor (associated with each hyperparameter sample ξ_i) $p(r^*|\mathbf{q}^*, \xi_i, \widetilde{\mathcal{D}}) = \mathcal{N}(\mu_{G_i}, \Sigma_{G_i})$ according to a Gaussian mixture model (GMM):

$$p(r^*|\mathbf{q}^*, \widetilde{\mathcal{D}}) = \sum_i \rho_i \mathcal{N}(\mu_{G_i}, \Sigma_{G_i}), \tag{25}$$

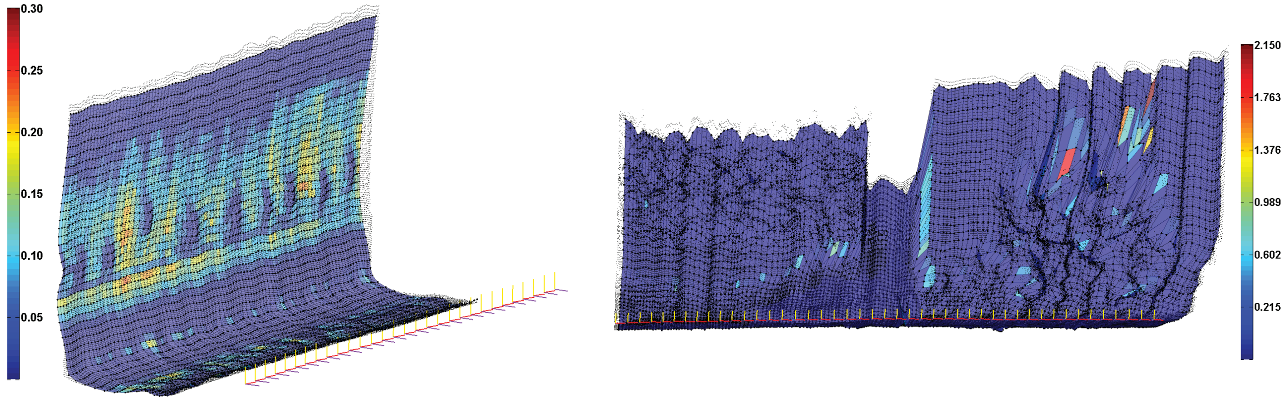
where the weightings ρ are subject to the constraint $\sum_i \rho_i = 1$. The mean and covariance of this summation of conditionally independent Gaussians are easily extracted:

$$\mu_{\mathrm{GMM}} = \sum_i \rho_i \mu_{G_i}, \tag{26}$$

$$\Sigma_{\mathrm{GMM}} = \sum_i \rho_i (\Sigma_{G_i} + \mu_{G_i}\mu_{G_i}^{\mathrm{T}}) - \mu_{\mathrm{GMM}}\mu_{\mathrm{GMM}}^{\mathrm{T}}. \tag{27}$$

Each of the weightings ρ_i are dependent on $p(\widetilde{\mathcal{D}}|\xi_i)$, assumed knowledge $p(\xi)$ and, importantly, the form of the predictive surfaces of both $p(r^*|\mathbf{q}^*, \xi, \widetilde{\mathcal{D}})$ and $p(\widetilde{\mathcal{D}}|\xi)$, as approximated by X and Y.

A strong feature of this approach is that the marginal likelihood $p(\widetilde{\mathcal{D}}|\xi_i)$ automatically assigns a high weighting to the GP regressors that explain the current active subset of measurements comparatively well, without the need for expert knowledge. As the active window moves across the workspace, the weightings of $p(r^*|\mathbf{q}^*, \xi_i, \widetilde{\mathcal{D}})$ freely adapt according to the changing workspace environment. Further, by ensuring that individual ξ_i are than recalculate



Max error: 0.26 m

Max error: 2.15 m

Fig. 11. Error in surface predictions conditioned on the actively selected predictive support window $\widetilde{\mathcal{D}}$ as the active window progressed across sensor configuration space $\mathcal{Q}$ (as per Section 6.3). 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. A profile view of a short section of (left) is also plotted in Figure 10. In the left part, where there is an absence of large discontinuities, we note that the smallest errors are near $\mathcal{D}'$ (dark blue areas). In the right part, there is a much more peaky distribution of error with significant errors appearing only where there are noisy surfaces and large discontinuities. 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 5.3.

each of the corresponding stationary GPs independently (Section 6.2), whilst producing an overall non-stationary regressor.

9.3. *Approximate Gaussian mixture model*

Although the additional quadrature parameters $\{p(\xi), X, Y\}$ can be estimated via a variety of optimization techniques as suggested in Osborne (2010), we offer two approximations that afford us large computational savings whilst maintaining an overall GMM approach. We argue that although it is essential to properly marginalize parameters ζ to ensure a flexible model, the hyperparameters ξ are far more forgiving in terms of the sensitivity of the resulting surface regression accuracy. The expensive calculation of Equation (24) can therefore be avoided by choosing to model average $p(r^*|\mathbf{q}^*, \xi, \widetilde{\mathcal{D}})$ through the use of $p(\widetilde{\mathcal{D}}|\xi)$ directly (without X and Y), and by embodying the equivalent of $p(\xi)$ implicitly through the positioning of ξ_i (see Section 9.4). The resulting computational cost is linear in the number of ξ_i. The algorithm is also trivial to parallelize as an independent GP regressor can be maintained for each ξ_i. We return to the application of full predictive Bayesian quadrature of Osborne (2010), discussing necessary considerations to manage the computational complexity without degradation of predictor performance, as future work (Section 12).

Concretely, we approximate the output of the GMM as a normal distribution which we use as input for our active sampling strategy (Section 5.3).

$$p(r^*|\mathbf{q}^*, \widetilde{\mathcal{D}}) \approx \mathcal{N}(\mu_{\mathrm{GMM}}, \Sigma_{\mathrm{GMM}})\,. \tag{28}$$

To form this prediction we maintain a bank of GP regressors with each element of the bank corresponding to a three-dimensional hyperparameter sample ($\xi_i = \{\sigma^2_{p_i}, l_{t_i}, l_{\theta_i}\}$). Every time a new laser measurement is accepted into the active subset we update all GP regressors according to Section 6.2. Each prediction $p(r^*|\mathbf{q}^*, \xi_i, \widetilde{\mathcal{D}}) = \mathcal{N}(\mu_{G_i}, \Sigma_{G_i})$ is combined into our overall predictive distribution through weighting with ρ_i (following Equation (25)) according to two alternative approximations.

9.3.1. *Weighting via normalized marginal likelihood*

$$\rho_i = \frac{p(\widetilde{\mathcal{D}}|\xi_i)}{\max p(\widetilde{\mathcal{D}}|\xi_i)}. \tag{29}$$

9.3.2. Maximum marginal likelihood estimate For comparison purposes we also choose the single hyperparameter sample ξ_{ML} that is the maximum marginal likelihood for the current predictive support window $\widetilde{\mathcal{D}}$:

$$p(r^*|\mathbf{q}^*, \widetilde{\mathcal{D}}) = p(r^*|\mathbf{q}^*, \xi_{\mathrm{ML}}, \widetilde{\mathcal{D}})\,, \tag{30}$$

$$\xi_{\mathrm{ML}} = \xi_i, \tag{31}$$

where

$$i = \underset{\mathrm{i}}{\mathrm{argmax}}\, p(\widetilde{\mathcal{D}}|\xi_i)\,. \tag{32}$$

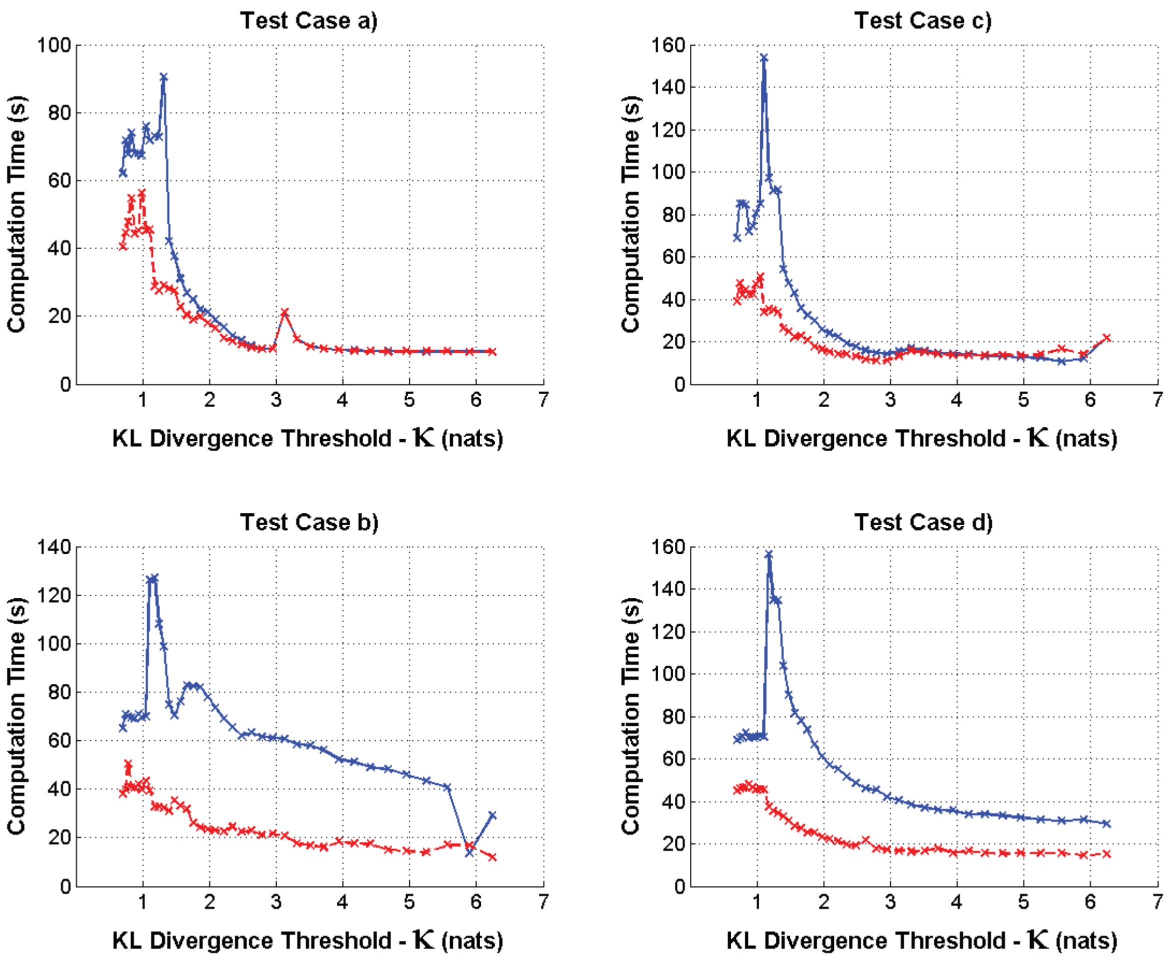


Fig. 12. Computation time versus prediction threshold κ for each of the test cases (a) to (d) (see Figure 6) using unoptimized MATLAB® code. Reported times exclude time spent predicting hold out measurements (which are only used only for the production of error metrics) using introspection (solid blue lines) and without using introspection (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.

9.4. *Choosing the quadrature prior* $p(\xi)$

Each of our approximations remove the dependency on the explicit prior term $p(\xi)$ in Equation (24). Instead we emulate this distribution through the positioning of ξ_i in our GMM. To demonstrate our approach, we use an uninformative $p(\xi)$, that of a bounded uniform prior. This is embodied through a regular grid of ξ_i. In addition to modifying ρ directly, expert knowledge can also be inserted by placing more ξ_i in regions where we require a greater weighting in the GMM. This knowledge may be available, for example, when the sinusoidal term of Section 9.1 is included in $k(\mathbf{q}, \mathbf{q}')$ and the typical range of vehicle periodicity is known. In the Bayesian quadrature framework of Osborne (2010) a variety of techniques to move individual ξ_i online according to the current fit of the regressor to the data are also suggested.

We aim to produce a high model flexibility by using a broad range of hyperparameter samples ξ_i. With each additional dimension of our hyperparameter sample, however, the curse of dimensionality becomes more apparent. We therefore use only a few ξ_i in each dimension to allow computations to remain manageable. In addition, 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 because ρ is dependent on a measure of the fit of our predictions to the training data ($p(\widetilde{\mathcal{D}}|\xi_i)$). Figure 15 demonstrates the outcome when very low length scales and high

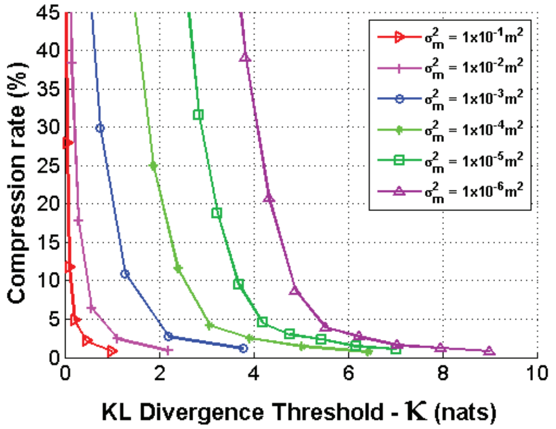


Fig. 13. Compression rate versus κ for test case (c) (see Figure 6) for varying measurement variance σ_m^2. Each line corresponds to a different measurement variance σ_m^2, which is used in our measurement model to form our likelihood $p(r^*|\zeta, \mathbf{q}^*, \xi, \mathcal{D})$ (Equation (5)). As 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 (18)). For comparison of associated error, see the box plots in Figures 9 and 14, where $\sigma_m^2 = 0.01$ m^2 and $\sigma_m^2 = 1 \times 10^{-4}$ m^2, respectively. From Sections 9 onwards we use $\sigma_m^2 = 1 \times 10^{-4}$m^2.

process variance are included in our set of hyperparameter samples $\{\xi_i\}$.

9.5. *Calculation of variance*

As with the stationary GP approach in Section 5, in practice we are only ever interested in the variance of range predictions and not their covariance. Equations (26) and (27) therefore collapse to

$$\mu_{\text{GMM}} = \rho^{\text{T}} \mu_G, \tag{33}$$

$$\sigma^2_{\text{GMM}} = \rho^{\text{T}}(\sigma_G^2 + \mu_G^2) - \mu^2_{\text{GMM}}, \tag{34}$$

where μ_G and σ_G^2 are vectors with each element corresponding to a prediction from each GP regressor with hyperparameter sample ξ_i.

10. Non-stationary results

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

10.1. *Comparing approximate GMMs*

Figure 16 shows compression rate $|\mathcal{D}'|/|\mathcal{D}|$ % across all test cases for our approximate GMM models.

Although the decrease in compression rate between the stationary (Figure 14) and non-stationary (Figure 16) model is stark, we must also take into account the resulting error for a fair comparison. In Table 1 we collate an exemplar of roughly equivalent compression rate for each test case from Figures 14 and 16. We see that results are similar across all regression models. As we saw in Figure 11 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 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 GMM approximation ($\rho_i = p(\widetilde{\mathcal{D}}|\xi_i)$) that uses all hyperparameter samples ξ_i 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 17 we plot η for both the stationary and the GMM model $\rho_i = p(\widetilde{\mathcal{D}}|\xi_i)$ approximation (the maximum marginal likelihood ξ_{ML} approximation is very similar). We also depict the hyperparameter samples that correspond to ξ_{ML}, which indicate the changing workspace complexity for each example. We see that the non-stationary η undulates more freely in response to changing workspace conditions, than the stationary one. 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 17 indicate the number of significant GMM weights ρ for each prediction. We see that for test case (d) there is seldom a region where more than one hyperparameter sample ξ_i is relevant to predictions. This provides an insight to why both non-stationary models achieve similar levels of surface reconstruction accuracy. In the following section we examine the spatial distribution of the errors to illustrate the effects where multiple ξ_i become relevant.

10.2. *Comparing surface error*

We can view the distribution of surface error in 3D Euclidean space to understand the correlation of error with

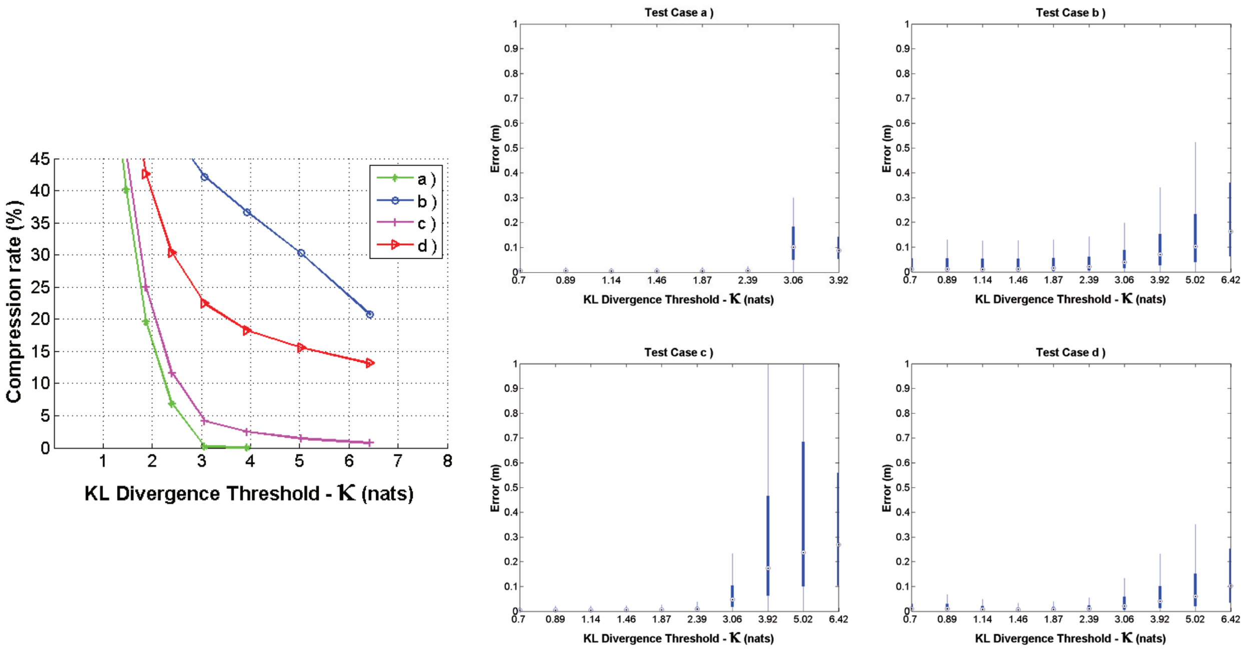


Fig. 14. Summary plots of the test cases in Figure 6 demonstrating the effect of varying κ for a measurement variance $\sigma_m^2 = 1 \times 10^{-4}$ m^2. These plots are provided for comparison with Figures 8 and 9 where $\sigma_m^2 = 0.01$ m^2. (Left) Compression rate $|\mathcal{D}'|/|\mathcal{D}|$ % versus κ. As the prediction threshold κ increases, compression rate decreases. By comparing with Figure 8 we note that when using a greater measurement variance for test case (b) that a lower compression rate is produced. 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, which may on average fit training data more effectively, and hence achieve a lower compression rate in Figure 8. Box plots (right) denote median and interquartile range for each of the κ indicated by markers in the plot on the left. Whiskers are plotted that extend a maximum of one and a half times the interquartile range.

local workspace complexity as we did in Figure 11 of Section 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 18. 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 17. Overall, the error in Figure 18 progressively decreases between plots from the left to the right.

11. Conclusions

In this paper we present a detailed account and an in-depth analysis of the adaptive 3D point-cloud compression algorithm first proposed by Smith et al. (2010a). The employed GP framework provides for a continuous representation of the implicit 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 squared in the size if of the active window, which is constant, rather than cubic in the size of the dataset.

The analysis presented in this paper indicates that the use of a stationary covariance function, as advocated in Smith et al. (2010a), results in almost no data compression in areas that include highly complex structures. Above and beyond the approach of Smith et al. (2010a), in this work we have investigated this property by enhancing our model through the use of non-stationary regressors. These are obtained by allowing the weighting on GP regressors with fixed hyperparameters to vary automatically according to the data. This allows the active window to adapt to workspace complexity and achieves a significant increase in performance over the stationary model. To enable this non-stationary behaviour, we extend the treatment of the regression problem in Smith et al. (2010a) by marginalizing over both model parameters and hyperparameters. Bayesian quadrature (O'Hagan 1991; Rasmussen and Ghahramani 2003; Osborne 2010) is used to approximate the intractable integrals. Two alternative approximations are also provided that further reduce computation requirements by

Table 1. Exemplars of results presented in Figures 14 and 16 of test cases (a) to (d) (see Figure 6). 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 $|\mathcal{D}'|/|\mathcal{D}|\,\%$ is roughly comparable. 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). When compared with the stationary model, both non-stationary models provide a clear increase in accuracy, even when the stationary model uses more measurements (test case (b)). Accounting for the slight increase in compression rate for the GMM $\rho_i = p(\widetilde{\mathcal{D}}|\xi_i)$ approximation, the two non-stationary models are very similar.

		Stationary			
Test case	$\kappa(nats)$	$\|\mathcal{D}'\|/\|\mathcal{D}\|\,\%$	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

		ξ_{ML} approximation				$\rho_i = p(\widetilde{\mathcal{D}}\|\xi_i)$ approximation			
Test Case	κ $(nats)$	$\|\mathcal{D}'\|/\|\mathcal{D}\|\,\%$	Median (m)	Mean (m)	Standard deviation (m^2)	$\|\mathcal{D}'\|/\|\mathcal{D}\|\,\%$	Median (m)	Mean (m)	Standard deviation (m^2)
(a)	1.145	7.53	0.00585	0.00734	0.00621	7.58	0.00587	0.00728	0.00597
(b)	3.069	35.3	0.0271	0.0653	0.112	35.3	0.0271	0.0653	0.112
(c)	3.069	4.68	0.0321	0.117	0.188	4.70	0.0309	0.108	0.169
(d)	3.069	18.3	0.0235	0.0779	0.206	18.4	0.0239	0.0768	0.205

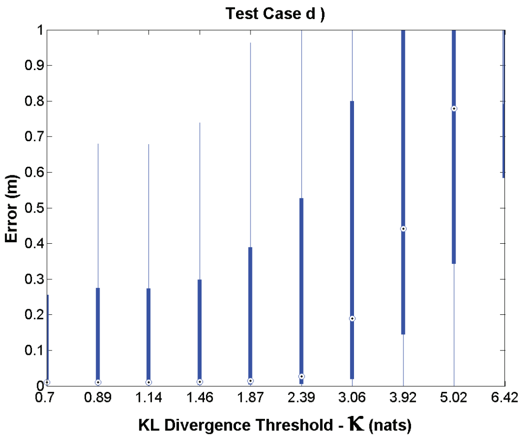


Fig. 15. Demonstration of the sensitivity of this approach to the range of hyperparameters samples ξ_i. Box plots denote median and interquartile range of error for a selection of logarithmically spaced thresholds κ for test case (d) (see Figure 6). Whiskers are plotted that extend a maximum of one and a half times the interquartile range. By allowing our set of hyperparameter samples $\{\xi_i\}$ to include very small length scales $\{l_t, l_\theta\} = \{0.1, 0.1\}$ units and a very high process variance $\sigma_p^2 = 0.7$ m^2 we produce measurement predictions that are highly overfitted. The resulting error statistics are poor compared to, for example, Figures 14 and 16.

exploiting the low sensitivity of our approach with respect to hyperparameter values. This allows the consideration of a large range of hyperparameters with a manageable computational cost. The resulting non-stationary algorithm is linear in com- plexity with the number of hyperparameter samples and is trivial to parallelize.

12. Future work

We wish to investigate further means to increase the compression of laser point clouds whilst also increasing the accuracy of surface predictions that are made using these compressed clouds. Bayesian quadrature techniques such as Osborne (2010) may provide significant improvements when used with a dense sampling in hyperparameter space. Alternatively the repositioning of individual hyperparameter samples according to changing workspace complexity, as per Osborne (2010), could place them so that more are relevant to each prediction. We can also employ derivative observations (Osborne 2010) to mitigate most conditioning issues associated with the additional quadrature parameters that are required in the resulting Bayesian quadrature approach. However, to ensure that the shape of the marginalization integrals are properly captured, and to guarantee that conditioning is never problematic, online assessment of the quadrature parameters may be required. A number of techniques are suggested in Osborne (2010) that are capable of managing this task.

Further gains in accuracy may arise from investigating the explicit basis functions used in our stochastic model. The use of a mixture of GPs, whereby each GP delineates

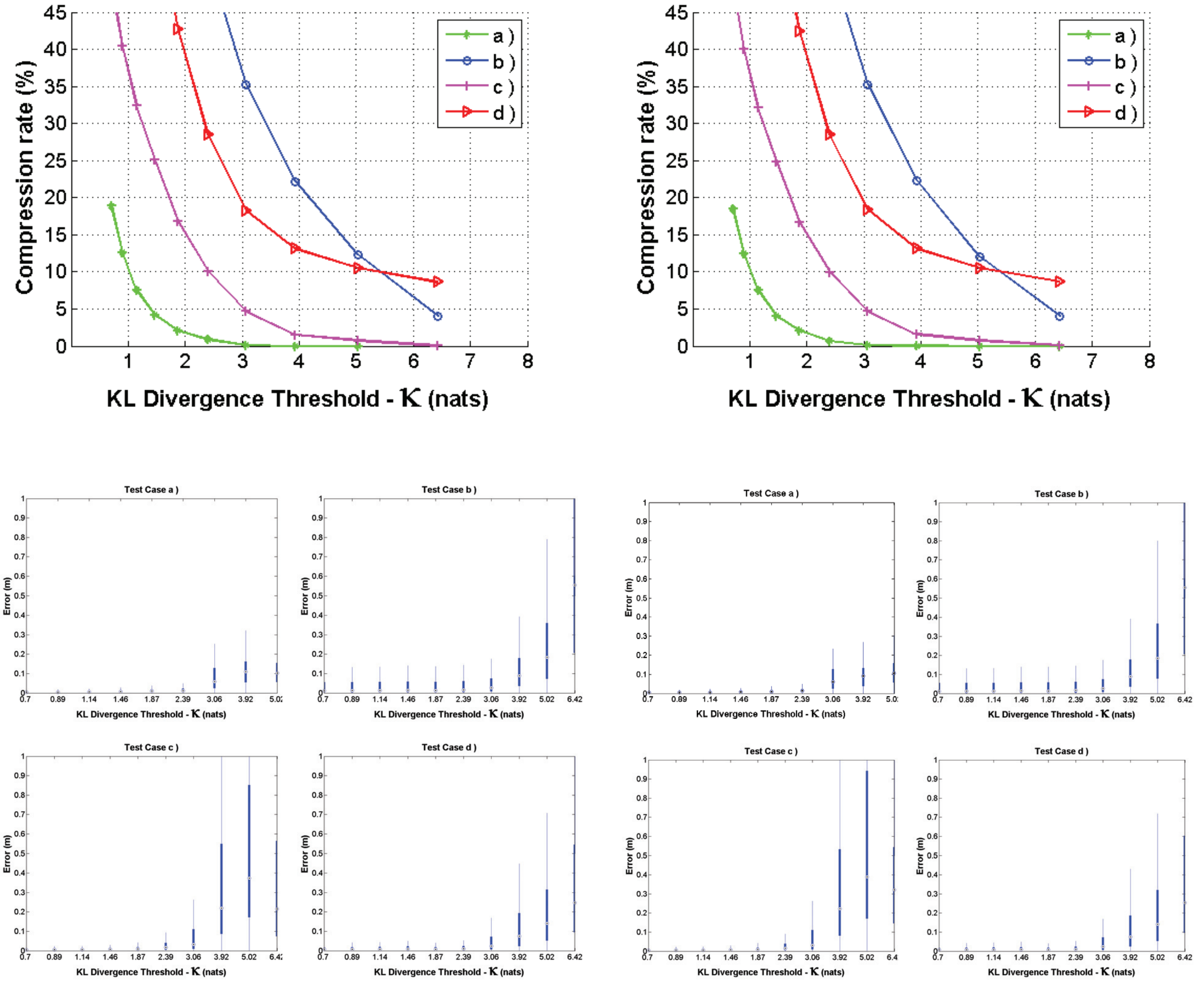


Fig. 16. Summary plots of results using our non-stationary regression models for the test cases (a)–(d) (see Figure 6). On the left are results for the maximum marginal likelihood ξ_{ML} approximation and, on the right, the GMM model $\rho_i = p(\widetilde{\mathcal{D}}|\xi_i)$. (Top row) Compression rate $|\mathcal{D}'|/|\mathcal{D}|$ % versus κ. For a given κ, compression rate is far lower for both these approximations than in the stationary model (Figure 14). 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 1.

a region of homogeneous workspace complexity, may also provide marked increases in accuracy surrounding large discontinuities. Unlike segmentation techniques that rely on expert knowledge (Smith et al. 2010b) we could use our measurement predictions to determine GP boundaries in a principled manner. This could afford outlier detection which so far has not been considered. Alternatively, increases in performance could arise from filtering outliers before they enter our GP framework.

As mentioned in Section 9.1, our approach permits the use of many combinations of stationary covariance functions. 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. Bayesian quadrature could be used to boost performance of this inference as it permits an *a posteriori* prediction of the most likely hyperparameters, rather than the discrete values that are produced by this current algorithm.

In workspaces where the same regions are repeatedly mapped, one can appreciate that tiny compression rates could be achieved if we re-used only the set of retained measurements that was selected on the first iteration. To do this, one would require a method to map between different sections of our sensor configuration space parameterization. This mapping could also provide a means to detect dynamic objects in the scene. We are also interested in

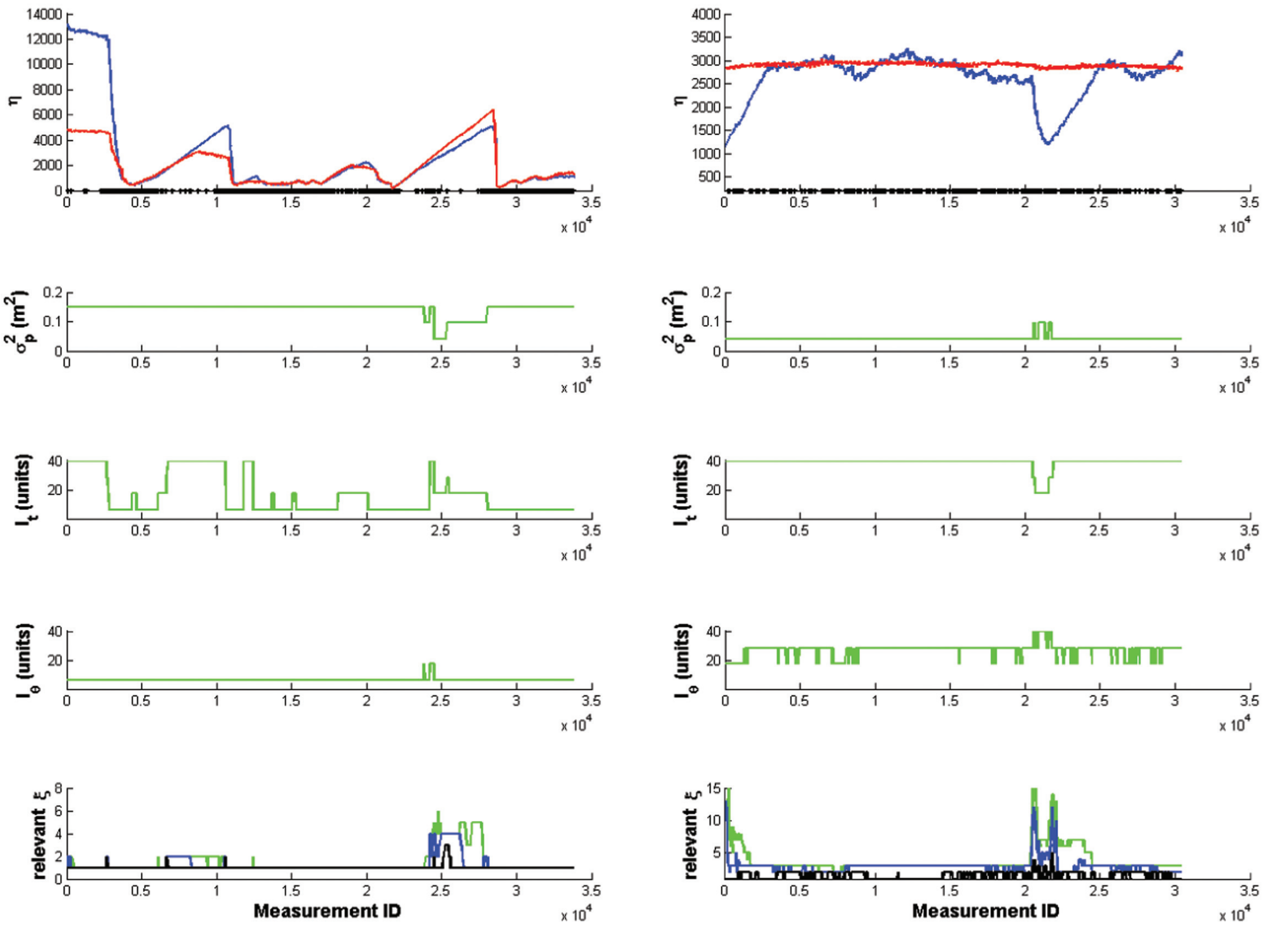


Fig. 17. Visualization of the active window as it moved across the sensor configuration space $\mathcal{Q}$. These plots correspond to the exemplars of test cases (a) and (d) in Table 1. Along the horizontal axes are the measurements as they are encountered by the active window as it progressed across $\mathcal{Q}$. 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 $\rho_i = p(\widetilde{\mathcal{D}}|\xi_i)$ 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 $\widetilde{D}$ as a result of introspection for the GMM approximation (see Section 5.3). (Middle three rows) Hyperparameters $\{\sigma_p^2, l_t, l_\theta\}$ that are used in the maximum marginal likelihood ξ_{ML} approximation are provided to indicate how the workspace complexity changed across the test cases. (Bottom row) Plot of the number of hyperparameter samples with weightings $\rho_i = p(\widetilde{\mathcal{D}}|\xi_i)$ that are significant compared to ξ_{ML}. The green line indicates the number of hyperparameter samples that have a relative weighting greater than $e^{-15} = 3.1 \times 10^{-7}$, blue $e^{-10} = 4.5 \times 10^{-5}$, and black $e^{-2} = 1.3 \times 10^{-1}$. 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 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.

online assessment of the size of the active set to reduce computational load where a small set will suffice.

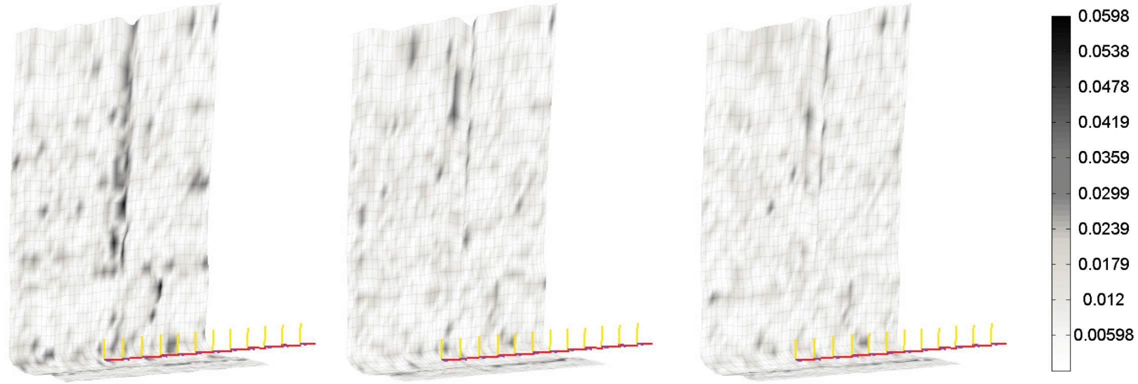
Notes

[1] Note that the regression will always be dependent on the ground-truth trajectory of the vehicle but that no knowledge of this trajectory is required in our algorithm.

[2] To convert this length scale into a value that is meaningful in terms of time and θ one must multiply by the resolution of the laser scanner ($\frac{1}{75}$ seconds and 0.5°, respectively).

Funding

This work was funded through an Engineering and Physical Sciences Research Council CASE award with BAE Systems, and by



Stationary
Max error: 0.0598 m

ξ_{ML} approximation
Max error: 0.0512 m

$\rho_i = p(\widetilde{\mathcal{D}}|\xi_i)$ approximation
Max error: 0.0452 m

Fig. 18. Error in surface predictions conditioned on the actively selected predictive support window $\widetilde{\mathcal{D}}$ for each of our regression models. Predictions are made online as discussed in Section 6.3 and projected into Euclidean x, y, z space via the trajectory of the sensor. 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 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. 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 $\rho_i = p(\widetilde{\mathcal{D}}|\xi_i)$ approximation, as detailed in Table 1. The decrease in maximum error when using the $\rho_i = p(\widetilde{\mathcal{D}}|\xi_i)$ approximation are however significantly less, and the distribution is slightly more even.

the European Commission (grant agreement number FP7-231888-EUROPA).

Acknowledgements

We would like to thank Winston Churchill for provision of the CAD models used throughout this paper, Mike Osborne, Rohan Paul and Ashley Napier for their helpful discussion, and the OeRC for use of their compute cluster to generate results.

References

Ansley C and Kohn R (1985) Estimation, filtering, and smoothing in state space models with incompletely specified initial conditions. *The Annals of Statistics* 13: 1286–1316.

Bares J, Hebert M, Kanade T, Krotkov E, Mitchell T, Simmons R, et al. (1989) Ambler: An autonomous rover for planetary exploration. *IEEE Computer* 22: 18–26.

Csató L and Opper M (2002) Sparse on-line Gaussian processes. *Neural Computation* 14: 641–668.

Deisenroth M, Rasmussen C and Peters J (2009) Gaussian process dynamic programming. *Neurocomputing* 72: 1508–1524.

Dongara J, Moler C, Bunch J and Stewart G (1979) *LINPACK users' Guide*. Philadelphia, PA: SIAM.

Früh C and Zakhor A (2003) Constructing 3D city models by merging aerial and ground views. *IEEE Computer Graphics Applications* 23: 52–61.

Hadsell R, Bagnell J, Huber D and Hebert M (2010) Non-stationary space-carving kernels for accurate rough terrain estimation. *The International Journal for Robotics Research* 29: 981–996.

Hähnel D, Burgard W and Thrun S (2003) Learning compact 3D models of indoor and outdoor environment with a mobile robot. *Robotics and Autonomous Systems* 44: 15–27.

Hoppe H, DeRose T, Duchamp T, McDonald J and Stuetzle W (1993) Mesh optimization. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*. New York: ACM Press, pp. 19–26.

Krause A, Singh A and Guestrin C (2008) Near-optimal sensor placements in Gaussian processes: theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research* 9: 235–284.

Lang T, Plagemann C and Burgard W (2007) Adaptive non-stationary kernel regression for terrain modeling. In *Robotics: Science and Systems (RSS)*.

MacKay D (2003) *Information Theory, Inference, and Learning Algorithms*. Cambridge: Cambridge University Press.

Martin MC and Moravec H (1996) *Robot Evidence Grids*. Technical Report CMU-RI-TR-96-06, Robotics Institute, Carnegie Mellon University.

Matheron G (1973) The intrinsic random functions and their applications. *Advances in Applied Probability* 5: 439–468.

Newman P, Sibley G, Smith M, Cummins M, Harrison A, Mei C, et al. (2009) Navigating, recognising and describing urban spaces with vision and laser. *The International Journal of Robotics Research* 28: 1406–1433.

Nguyen-Tuong D and Peters J (2010) Using model knowledge for learning inverse dynamics. In *IEEE International Conference on Robotics and Automation*.

O'Hagan A (1987) Monte Carlo is fundamentally unsound. *The Statistician* 36: 247–249.

O'Hagan A (1991) Bayes–Hermite quadrature. *Journal of Statistical Planning and Inference* 29: 245–260.

O'Hagan A and Kingman J (1978) Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)* 40: 1–42.

Osborne M (2010) *Bayesian Gaussian Processes for Sequential Prediction, Optimisation and Quadrature*. PhD thesis, University of Oxford.

Osborne M, Roberts S, Rogers A, Ramchurn S and Jennings N (2008) 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*. Washington, DC: IEEE Computer Society, pp. 109–120.

Pfaff P and Burgard W (2005) An efficient extension of elevation maps for outdoor terrain mapping. In *International Conference on Field and Service Robotics (FSR)*.

Plagemann C, Kersting K, Pfaff P and Burgard W (2007) Gaussian beam processes: a nonparametric Bayesian measurement model for range finders. In *Robotics: Science and Systems (RSS)*.

Plagemann C, Mischke S, Prentice S, Kersting K, Roy N and Burgard W (2008) Learning predictive terrain models for legged robot locomotion. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Quiñonero-Candela J and Rasmussen C (2005) A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research* 6: 1939–1959.

Rasmussen C and Ghahramani Z (2003) Bayesian Monte Carlo. In *Advances in Neural Information Processing Systems*, pp. 505–512.

Rasmussen CE and Williams CKI (2006) *Gaussian Processes for Machine Learning*. Cambridge, MA: The MIT Press.

Rusinkiewicz S and Levoy M (2001) Efficient variants of the ICP algorithm. In *Proceedings of 3DIM*, pp. 145–152.

Scholkopf B and Smola A (2002) *Learning with Kernels*. Cambridge, MA: The MIT Press. http://www.learning-with-kernels.org/.

Seeger M (2008) *Low Rank Updates for the Cholesky Decomposition*. Technical Report, University of California at Berkeley.

Seeger M, Williams C and Lawrence N (2003) Fast forward selection to speed up sparse Gaussian process regression. In *Workshop on AI and Statistics*, volume 9.

Smith M, Baldwin I, Churchill W, Paul R and Newman P (2009) The new college vision and laser data set. *The International Journal of Robotics Research* 28: 595–599.

Smith M, Posner I and Newman P (2010a). Efficient non-parametric surface representations using active sampling for push broom laser data. In *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain.

Smith M, Posner I and Newman P (2010b) Generating implicit surfaces from lidar data. In *Towards Autonomous Robotic Systems*, Plymouth, UK.

Stein ML (1999) *Interpolating Spatial Data*. Berlin: Springer-Verlag.

Triebel R, Burgard W and Dellaert F (2005) Using hierarchical EM to extract planes from 3D range scans. In *International Conference on Robotics and Automation (ICRA)*.

Triebel R, Pfaff P and Burgard W (2006) Multi-level surface maps for outdoor terrain mapping and loop closing. In *International Conference on Intelligent Robots and Systems (IROS)*.

Vasudevan S, Ramos F, Nettleton E, Durrant-Whyte H and Blair A (2009) Gaussian process modeling of large scale terrain. In *International Conference on Robotics and Automation (ICRA)*.

Appendix: Marginal likelihood over function values

To weight each of the stationary GP regressors in our non-stationary GMM, we use a measure of the efficacy of each GP in predicting the current active set of observations $\mathbf{r}$. Following the Bayesian framework we turn to the marginal likelihood with respect to $\mathbf{r}$, which for $Z(\mathbf{q}) \sim \mathcal{GP}(0, k(\mathbf{q}, \mathbf{q}'))$ of Section 5 is simply

$$\log p(\widetilde{\mathcal{D}}|\xi)_Z = -\frac{1}{2}\Big[\mathbf{r}^{\mathrm{T}} K_{\mathbf{r}}^{-1}\mathbf{r} - n\log 2\pi \\ - \log|K_{\mathbf{r}}|\,\Big]. \tag{35}$$

The calculation the marginal likelihood for the stochastic mean GP of Section 5.2 is also standard as shown in Ansley and Kohn (1985)

$$\log p(\widetilde{\mathcal{D}}|\xi)_G = -\frac{1}{2}\Big[\mathbf{r}^{\mathrm{T}} K_{\mathbf{r}}^{-1}\mathbf{r} - (n-m)\log 2\pi \\ - \log|K_{\mathbf{r}}| - \log|A| + S\,\Big], \tag{36}$$

where

$$S = \mathbf{r}^{\mathrm{T}} K_{\mathbf{r}}^{-1} H^{\mathrm{T}} A^{-1} H K_{\mathbf{r}}^{-1}\mathbf{r} \tag{37}$$

and m is the rank of H.