Efficient Non-Parametric Surface Representations Using Active Sampling for Push Broom Laser Data

Mike Smith and Ingmar Posner and Paul Newman
Oxford University, Mobile Robotics Group
{mike, hip, pnewman} @ robots.ox.ac.uk

Abstract—**This paper concerns the creation of efficient surface representations from laser point clouds. We produce a continuous, implicit, non-parametric representation with an update time that is constant. The algorithm places no restriction on the complexity of the underlying workspace surfaces and automatically prunes redundant data via an information theoretic criterion. This criterion makes the use of GP 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 level and reconstruction error. We attain decimation factors in excess of two orders of magnitude without significant degradation in fidelity.**

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

I. Introduction and Motivation

3D point clouds gathered from outdoor scenes are now ubiquitous. While visually appealing when rendered, they have limitations; two of which are addressed in this paper. Firstly, they are discrete samples – they do not readily admit access to the continuous nature of the workspace's surfaces. Secondly, they are inefficient in the sense of containing samples which, given other nearby samples and reasonable smoothness assumptions, are redundant and convey little 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 data set retaining only locally salient samples. We attain 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 itself (using a sensor model) and its GP-predicted distribution, we arrive at a criterion for keeping or dropping individual samples. Here we draw inspiration from work in sparse GPs [20], [2], [25]. We refer to this process as Active Sampling – the
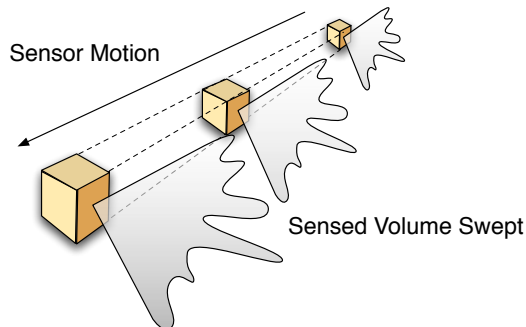
GP framework actively guides us in deciding which of many input points should be retained and later used for surface reconstructions.

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

II. 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, [5], [7]) 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 a workspace and/or coverage is incomplete – both are frequently the case.

An alternative approach considers a parametric representation based on geometric primitives [6], [28]. Planar representations are a popular choice since they are ubiquitous in
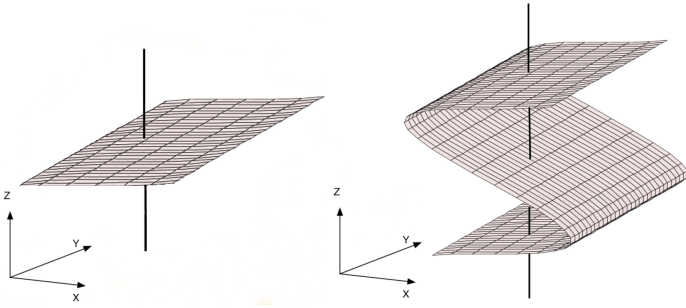
Figure 2. We wish to map scenes with multiple elevations for a given x, y point (right), rather than being restricted to a trivial case (left).

man-made environments. However, the reliance on geometric primitives places strong assumptions on the environment 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 [10], [1], [17], [29], which represent the world as a regular grid of cells or voxels. Each cell stores the probability of 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 [9], [19], [30], although they were first introduced to the geostatistics field under the name *kriging* many years previously by [11] amongst others. While these approaches differ in their choice of (non-stationary) covariance function and GP sparsification method, they adopt the same parameterisation of the problem – that is they model a function $f : \mathbb{R}^2 \to \mathbb{R}$ which associates a single elevation value z with any given position (x, y) in 3D Euclidean space. 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 parameterisation 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, with an overview provided in [20]. Within this field, active subsampling strategies have been used to select information-rich data through use of information theoretic criteria [8], [3], [25]. Our work is particularly similar to [25], but we exploit the time-sequential nature of laser data (see Section IV-B) 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, this work

bears closest relation to that of Gaussian Beam Processes [18]. The authors model laser data on a per-scan basis using a GP model to regress the 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.

III. NON-FUNCTIONAL SURFACE REPRESENTATION VIA BEAM-SPACE PARAMETERISATION

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 parameterised as a point $\mathbf{q} \in \mathbb{S} \times \mathbb{R}$ where $\mathbf{q} = [\theta, t]^T$ where 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 *beam space, $\mathcal{Q}$*, as illustrated in Figure 3 and refer to it as a particular *configuration* of the laser sensor. Beam 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 beam 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

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

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

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

where:

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

For every point $\mathbf{q}$ in beam space, $V(\mathbf{q})$ provides the six d.o.f. pose of the sensor $\mathbf{p} \in \mathbb{S}^3 \times \mathbb{R}^3$ at the time a measurement is taken – represented as roll, pitch, yaw and position. $E(\mathbf{p}, r)$ maps a six d.o.f sensor pose $\mathbf{p}$ and a scalar range measurement r to a single point $\mathbf{x}$ in 3D Euclidean space. This parameterisation naturally eschews the non-functional relation between elevation z and (x, y) location that is commonly found in terrain mapping formulations (Figure 2). By keeping each operation distinct, we also decouple robot trajectory estimation $V(\mathbf{q})$[1], from that of the regression of the laser data $G(\mathbf{q})$. This permits, for example, 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 postion in $\mathcal{Q}$ we turn to Gaussian Process Regression. Given a set of measurements $\mathcal{D} = \{(\mathbf{q}_i, r_i)\}_{i=1}^N$ at a query point $\mathbf{q}^*$ we can

---

[1]Note, however, the regression is dependent on the actual ground-truth trajectory of the vehicle.
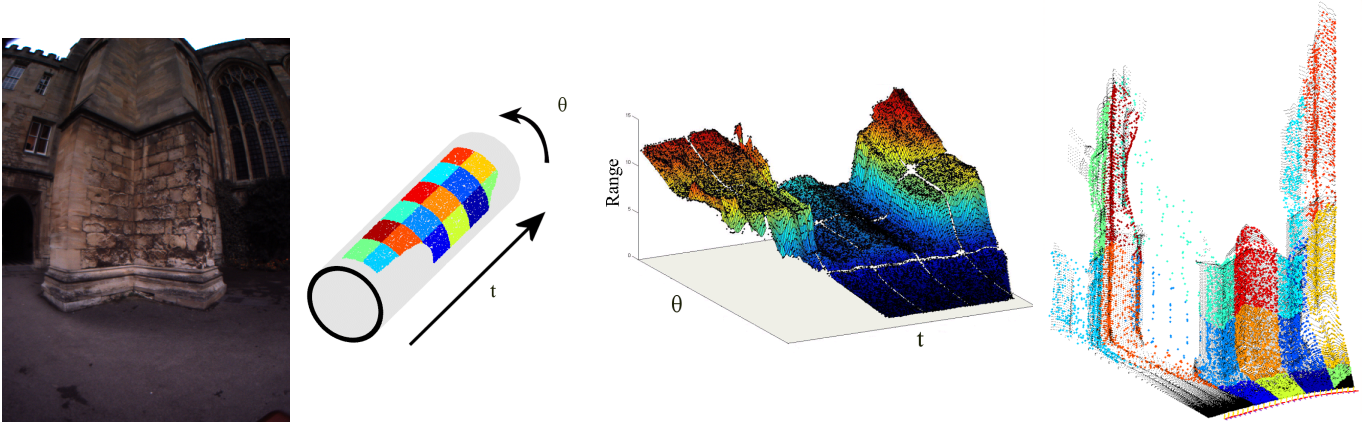
Figure 3. Parameterisation used in this work. (Left) A camera image used to provide the visual context for this example. The image is captured at the same time as the laser data plots in (left-mid) to (right), which forms the central part of exemplar d) in Figure 8 (see Section V). (Mid-left) Beam space manifold $\mathcal{Q}$. Every point $\mathbf{q}$ on this manifold is parameterised by time t and the angular position of the laser beam θ. (Mid-right) Laser range image. $\mathcal{Q}$ is unwrapped with t and θ of along the X and Y axes and output of the mapping $G(\mathbf{q})$, range predictions $\mathbf{r}$ along Z. (Right) 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 beam-space and euclidean space.

obtain a predictive distribution $p(r^*|\mathbf{q}^*, \mathcal{D})$. In the following section we present a brief summary of how this predictive distribution is obtained from a sliding window in beam-space.

IV. ACTIVE SLIDING WINDOW USING GAUSSIAN PROCESS REGRESSION

Gaussian Processes (GPs) provide for non-parametric probabilistic regression over $\mathcal{Q}$. A GP consists of a collection of jointly Gaussian distributed random variables and describes a distribution over latent functions underlying observations. It 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 $G(\mathbf{q})$ corresponding to these latent functions. Given known ranges $\mathbf{r}$ from different configurations $Q = \{\mathbf{q}_i\}_{i=1}^N$ and a query point $\mathbf{q}^*$ with corresponding unknown target range r^* we can write:

$$\begin{bmatrix} \mathbf{r} \\ r_* \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \mu(Q) \\ \mu(\mathbf{q}*) \end{bmatrix}, \begin{bmatrix} K(Q,Q) + \sigma_m^2 I & \mathbf{k}(Q, \mathbf{q}_*) \\ \mathbf{k}(Q, \mathbf{q}_*)^T & k(\mathbf{q}_*, \mathbf{q}_*) + \sigma_m^2 \end{bmatrix} \right) \tag{5}$$

Each element of K is, in the stationary case, only on a suitably chosen distance metric $d = \|\mathbf{q} - \mathbf{q}'\|$ between two corresponding points $\mathbf{q}$ and $\mathbf{q}'$ in beam space. We explicitly account for noise in the training observations $\mathbf{r}$ through an additive white noise process of strength σ_m along the diagonal entries of K^2. The derivation of the mean $\mathbb{E}[r_*]$ and covariance $\mathbb{V}[r_*]$ of the predictive distribution $p(r^*|\mathbf{q}^*, \mathcal{D})$ for a deterministic $\mu(\mathbf{q}) = 0$ (as is commonly used [30]) are standard and can be found, for example, in [21]

$$\mathbb{E}[r_*] = \mathbf{k}(Q, \mathbf{q}_*)^T (K(Q,Q) + \sigma_m^2 I)^{-1} \mathbf{r} \tag{6}$$

$$\mathbb{V}[r_*] = k(\mathbf{q}_*, \mathbf{q}_*) + \sigma_m^2 - \\ \mathbf{k}(Q, \mathbf{q}_*)^T (K(Q,Q) + \sigma_m^2 I)^{-1} \mathbf{k}(Q, \mathbf{q}_*) \tag{7}$$

^{2}We construct $\sigma_m^2 = 0.02$ from our sensor model in accordance with the precision of a SICK LMS291-S14 (15mm) and lab measurements.

Throughout this work we use a member of the Matérn class of covariance functions as advocated in [27]. We note, however, that many others, including non-stationary covariance functions [9] could be adapted and substituted in its place (Section VI). The Matérn class 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 in [21], 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 for a variety of support set sizes and length scales. 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

$$k(\mathbf{q}, \mathbf{q}')_{matern\ \nu=\frac{3}{2}} = \sigma_p^2 (1 + \frac{d\sqrt{3}}{l}) \exp(-\frac{d\sqrt{3}}{l}) \tag{8}$$

where σ_p^2 is the process noise, l is the length scale and $d = \|\mathbf{q} - \mathbf{q}'\|_{\mathcal{Q}}$ denotes the geodesic distance between $\mathbf{q}$ and $\mathbf{q}'$ in beam space, which for this sensor configuration is the ℓ_2 norm.

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 $\mathcal{D}$. Instead, for each prediction we enforce a fixed predictive support window size n, formed from the closest (in terms of the ℓ_1 norm from t to t_*) points that have been actively accepted by our algorithm to guarantee 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 it's trajectory.

We are interested in minimising the computational complexity of our algorithm for practical applications. In Section IV-B we devise an active sampling strategy to intelligently

determine a salient subset of $\mathcal{D}$, $\mathcal{D}'$ to include in our active window. The advantages of this are two-fold: we can achieve a significant compression of $\mathcal{D}$, and we can re-use the inversion of Equations 6 and 7 across multiple predictions, thus decreasing computational cost. Where we must incorporate new points into the active window we update, rather than re-calculate the inversion (Section IV-C). This lowers the overall worst case prediction cost to $\mathcal{O}(n^2)$.

A. Stochastic Mean Function

We affect a stochastic non-zero mean function $\mu(\mathbf{q})$ through the use of explicit basis functions [21]. Explicit basis functions $\mathbf{h}(\mathbf{q})$ with weightings $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{b}, B)$, that are found in a traditional Bayesian weight space view of linear regression, are superimposed with the zero-mean GP $Z(\mathbf{q}) \sim \mathcal{GP}(0, k(\mathbf{q}, \mathbf{q}'))$ of Section IV

$$G(\mathbf{q}) = Z(\mathbf{q}) + \mathbf{h}(\mathbf{q})^T \boldsymbol{\beta} \qquad (9)$$

Crucially, in the absence of data, predictions using $G(\mathbf{q})$ tend to the explicit basis function surface estimate, rather than that of the zero mean. One may view this formulation as the GP modelling the residuals between the implicit surface generated by the explicit basis function, and $\mathcal{D}$, and has been used successfully in robotics applications such as [14]. This is particularly beneficial when predicting at the boundary of our active window, and is the key to our compression rates (Section IV-B).

We take the limiting case as the prior on the weightings tends towards a uniform distribution, $(B^{-1} \to \mathbf{0})$. Following [15] we arrive at a predictive distribution that is independent of $\mathbf{b}$.

$$
\begin{aligned}
\mu_g(\mathbf{q}*) &= \mathbb{E}_Z[r*] + \mathbf{u}^T \bar{\boldsymbol{\beta}} & (10)\\
\sigma_g^2(\mathbf{q}*) &= \mathbb{V}_Z[r*] + \mathbf{u}^T (HK(Q,Q)^{-1}H^T)^{-1}\mathbf{u} & (11)
\end{aligned}
$$

where $\mathbb{E}_Z[r*]$ and $\mathbb{V}_Z[r*]$ are given by Equation 6 and 7 respectively, $H = [\mathbf{h}(\mathbf{q}_1), \mathbf{h}(\mathbf{q}_2)...\mathbf{h}(\mathbf{q}_n)]$ and

$$
\begin{aligned}
\bar{\boldsymbol{\beta}} &= (HK(Q,Q)^{-1}H^T)^{-1}HK(Q,Q)^{-1}\mathbf{r} & (12)\\
\mathbf{u} &= \mathbf{h}(\mathbf{q}_*) - HK(Q,Q)^{-1}\mathbf{k}(Q,\mathbf{q}*) & (13)
\end{aligned}
$$

This formulation affords us the benefits of explicit basis functions without requiring additional parameters, and maintains the desirable predictive properties of GPs. Like Equation 6 and 7 an inverse must be calculated, but its dimension is the number of explicit basis functions chosen for the application. We choose the polynomial set of basis functions $\mathbf{h} = [1, \theta, \theta^2, \theta^3, t, t^2, t^3, \theta t, \theta t^2, \theta^2 t, \theta^2 t^2]^T$ and have found cross terms above second order to have marginal overall impact on predictions.

B. Active Sampling through KL Divergence

In this Section we describe our method for active point selection. We decide if our prediction $r* \sim \mathcal{N}(\mu_g, \sigma_g^2)$ (Equation 10 and 11) differs significantly from each 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 if the average additional information required to specify r_m as a result of using $r*$ (instead of the true r_m) is greater than a threshold κ. This threshold, naturally modulated by terrain complexity, sets decimation fraction $\frac{|\mathcal{D}|}{|\mathcal{D}'|}$. Thorough analysis on parameter choice can be found in Section V. The KL divergence between two 1D Gaussian distributions has a closed form

$$D_{KL}(\mathcal{N}_m || \mathcal{N}_g) = \frac{1}{2}\left[log_e \frac{\sigma_g^2}{\sigma_m^2} + \frac{\mu_g^2 + \mu_m^2 - 2\mu_g\mu_m + \sigma_m^2}{\sigma_g^2} - 1\right] \qquad (14)$$

On adoption of any new point into the active support set we introspectively analyse the closest (ℓ_1 norm in $\mathcal{Q}$) r_m that had previously been decided redundant to determine if our belief has changed, given the new information. On adoption of this point introspection is then 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 sections of the workspace, our algorithm performs as well as a naïve implementation by using all $\mathcal{D}$, while maintaining the ability to automatically subsample simple scenes. We note 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 points can increase accuracy and robustness [22].

C. Implementation Considerations

We seek to maximise 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. We use Cholesky decomposition and rank one updates of K to perform the expensive inversion in Equations 6 and 7, as is common in the GP literature [16], [24], [23]. Each time we accept a new point into our active set we first marginalise the most stale point (at the back of the active window), then expand the active set to include the new measurement.

Marginalisation: Given a positive definite matrix K, and its Cholesky factor C such that $C^T C = K$

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

we can remove the central row and column by marginalising C to:

$$\begin{bmatrix} C_{1,1} & C_{1,3} \\ \mathbf{0} & \Upsilon(C_{3,3}^T C_{3,3} + \mathbf{c}_{2,3}^T \mathbf{c}_{2,3}) \end{bmatrix} \qquad (15)$$

Where Υ is the Cholesky update of [4] which exploits the special structure of $\mathbf{c}_{2,3}^T \mathbf{c}_{2,3}$, and is of complexity $O(n^2)$.
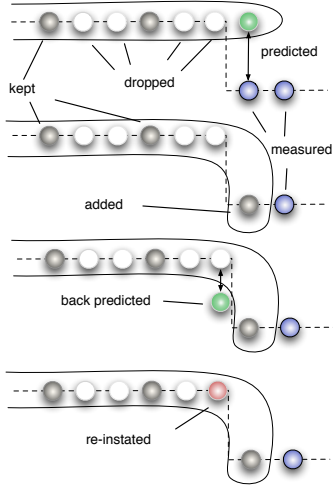
Figure 4. Active sampling of a 1D data series. Each node connected by the chained lines represents a measurement. Black indicates data accepted by the forward pass of the active sampling algorithm, while the white have been rejected and the blue are yet to be observed. Green nodes are predictions made by our algorithm and red are points 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.

An efficient implementation can be found in, for example, *cholupdate* of MATLAB®[12].

Expansion: Given $K = \begin{bmatrix} K_{1,1} & K_{1,3} \\ K_{1,3}^T & K_{3,3} \end{bmatrix}$ and corresponding $R = \begin{bmatrix} R_{1,1} & R_{1,3} \\ R_{1,3}^T & R_{3,3} \end{bmatrix}$, we wish to expand to C through insertion of a central row and column:

$$\begin{bmatrix} R_{1,1} & R_{1,1}^T \backslash \mathbf{k}_{1,2} & R_{1,3} \\ \mathbf{0} & \sqrt{k_{2,2} - \mathbf{c}_{1,2}^T \mathbf{c}_{1,2}} & \frac{\mathbf{k}_{2,3} - \mathbf{c}_{1,2}^T C_{1,3}}{c_{2,2}} \\ \mathbf{0} & \mathbf{0} & \Upsilon(R_{3,3}^T R_{3,3} - \mathbf{c}_{2,3}^T \mathbf{c}_{2,3}) \end{bmatrix}$$ (16)

where we have used $\backslash$ above to indicate the solution of $\mathbf{c}_{1,2}$ in the Equation $R_{1,1}^T \mathbf{c}_{1,2} = \mathbf{k}_{1,2}$ is found by the use of backwards or forwards substitution for the triangular $R_{1,1}$.

V. EXPERIMENTAL RESULTS

In this section we analyse how changing κ affects the decimation fraction of the original point cloud for data collected in a commonplace urban environment. Examples were obtained using the New College data set [26]. 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. Pose estimates were obtained using visual odometry from a forward facing stereoscopic camera [13]. Throughout we use the Matérn class of covariance functions, a length scale l of 8, a process variance σ_p^2 of $0.05m$ and a active window size of 200, all of which were found empirically
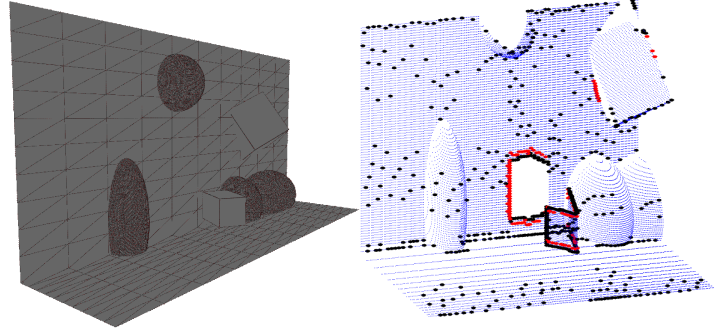

Figure 5. Result from using active sampling within our beam space formulation. (Left) CAD model used to generate synthetic data based on the wall section ($7 \times 3 \times 1.5m$) of exemplar a)(see Figure 8). (Right) Plot of synthetic data as a laser is dragged from left to right in front of the scene. Blue indicates all laser measurements, black denotes points selected on the forward wave of the algorithm, and red are points that have been chosen introspectively. We have used $\kappa = 0.8\,nats$ as in other examples (see Figure 8). Note that the floating cube in (Left) generates a large ($1m$) discontinuity which is automatically highly sampled on both sides. For a plot indicating the effect of varying κ see Figure 7.

to produce consistently accurate surface representations, for a given compression ratio, as discussed in Section IV. Error (in m) is defined as the ℓ_1 norm in range between a hold out set of measurements and corresponding predictions at the same point in $\mathcal{Q}$. Predictions are made as the centre of the active window passes the hold out set in $\mathcal{Q}$ and are conditioned on the current active set.

A. Compression

Figure 6 depicts typical subsampling as κ is varied. For aggressive compression factors of 1000 we achieve a modest mean error of $0.3m$. For errors comparable with measurement precision of $0.015m$ we can achieve a compression factor of 6.

In Figure 7 results are collated across a range of κ for the four exemplars, a) through d), in Figure 8 and the two synthetic cases, e) and f) (Figure 5). Intuitively, as the threshold is increased an increasing compression ratio is produced, accompanied by an increase in error as depicted by the box plots in Figure 8. As one can discern from the images and CAD models, the relative positions of the curves correspond to the scene complexity: the more complex the scene, the lower the compression ratio. Scenes b) and d) are the most complex, with noisy foliage, measurements of ceilings behind window panes and discontinuities as great as $5m$, compared to that of the $1m$ discontinuity the person in c) presents. In all cases, these complex regions have been sampled most heavily – the outline of the person can be recognised in the Euclidean plot of c) in Figure 8.

B. Surface Error

Box plots indicate surface error distributions for each of the examples in Figure 8. As compression increases, error typically migrates from being that of a tightly bunched distribution around that of the measurement noise, to one with an increased mean and a significant tail – into the metre range in the worst
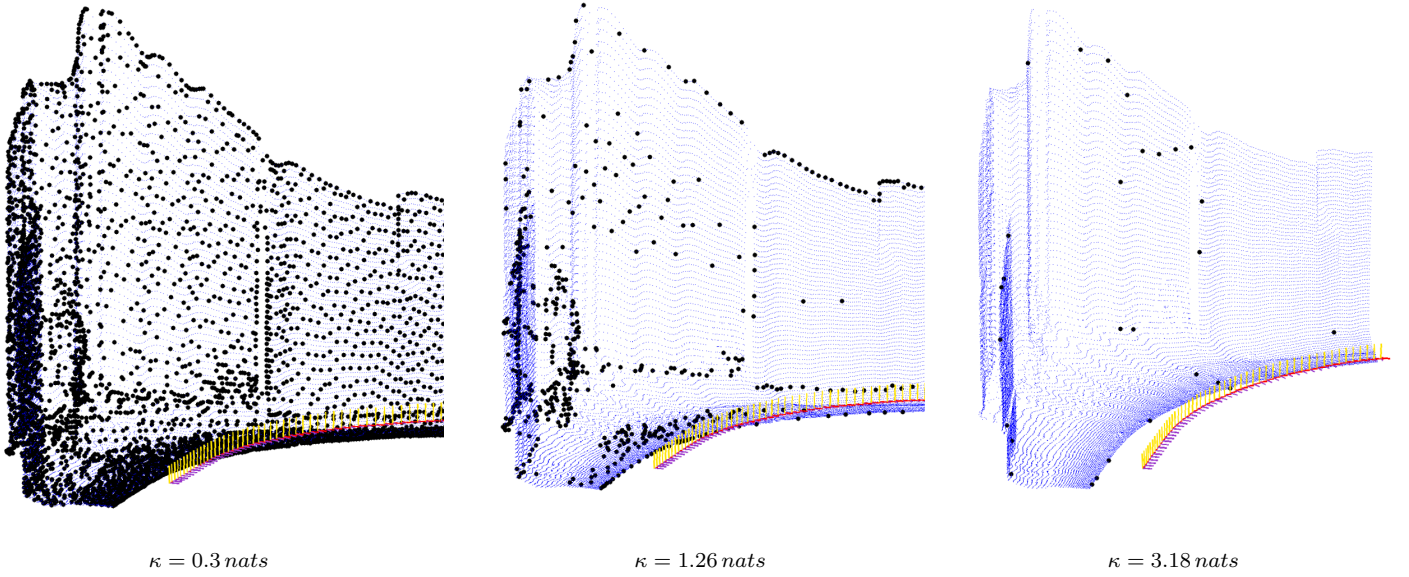
$\kappa = 0.3\,nats$ $\kappa = 1.26\,nats$ $\kappa = 3.18\,nats$

Figure 6. Point subsample plots for $\kappa = \{0.3, 1.26, 3.18\}\,nats$ for example c) of Figure 8. The 28892 Original measurements are in blue and points selected by our active sampling algorithm are in black. The vehicle trajectory is represented by the coloured frames across the bottom of each plot. (Left) 4543 laser measurements are used to produce predictions resulting in a mean error of $0.014m$ while in (middle) 323 points are used with a $0.177m$ average error. (Right) only 28 points have been selected, providing an average error of $0.32m$.
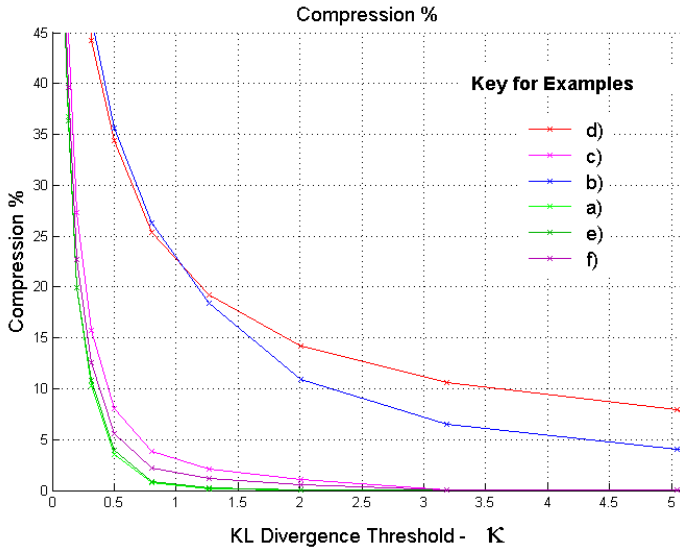

Figure 7. compression verses κ for a variety of urban environments. Images of the workspaces can be found in Figure 8 for examples a) through d) and a CAD model f) in Figure 5. Example e) is the wall and ground plane of example f), which is a simplified $1:1$ model of example a). Compression, measured as the percentage of $\mathcal{D}$ selected by our algorithm, is plotted against κ at each of the configurations denoted by crosses. Greater compression results from higher thresholds and scenes that are less complex.

$\kappa = 0.32\,nats$ $\kappa = 0.8\,nats$ $\kappa = 2\,nats$

Figure 9. Profile view of a short section of wall and ground plane from example a) (Figure 8 and 10), with $\kappa = \{0.32, 0.8, 2\}\,nats$ from left to right. $\mathcal{D}$ is shown in black, while predictions conditioned on $\mathcal{D}'$ are collated into the surfaces. Larger thresholds create larger point decimation, hence smoother surfaces that are in greater error.

implementation that uses all $\mathcal{D}$ as discussed in Section IV-B.

VI. CONCLUSIONS

In this paper we introduced a novel, beam space, parameterisation for push broom lasers. This formulation allows us to leverage well-formed techniques on what is in essence a non-functional problem domain. A Gaussian Process framework permits us a continuous surface representation using only a small subset of available points. Indeed, we can comfortably decimate point clouds of simple workspaces by factors in excess of two orders of magnitude, without significant degradation in fidelity. This is possible through use of a principled sampling strategy and a stochastic mean function. Further, we manage variables that are used throughout the active sampling process to ensure the algorithm is $\mathcal{O}(n^2)$ in the size if of the active window, which is constant, rather than $\mathcal{O}(N^3)$ in the size of the dataset.

case. A profile view of example a) is used to highlight how errors change with κ in Figure 9.

To demonstrate the spatial distribution of error we provide the surface predictions of example a) and b) coloured according to error in Figure 10. We observe that the worst errors results at discontinuities and very noisy surfaces such as foliage. In these areas the performance is that of a naïve
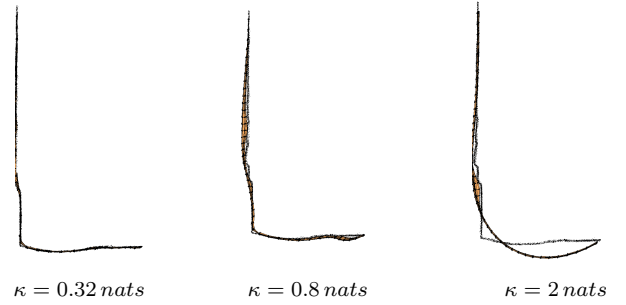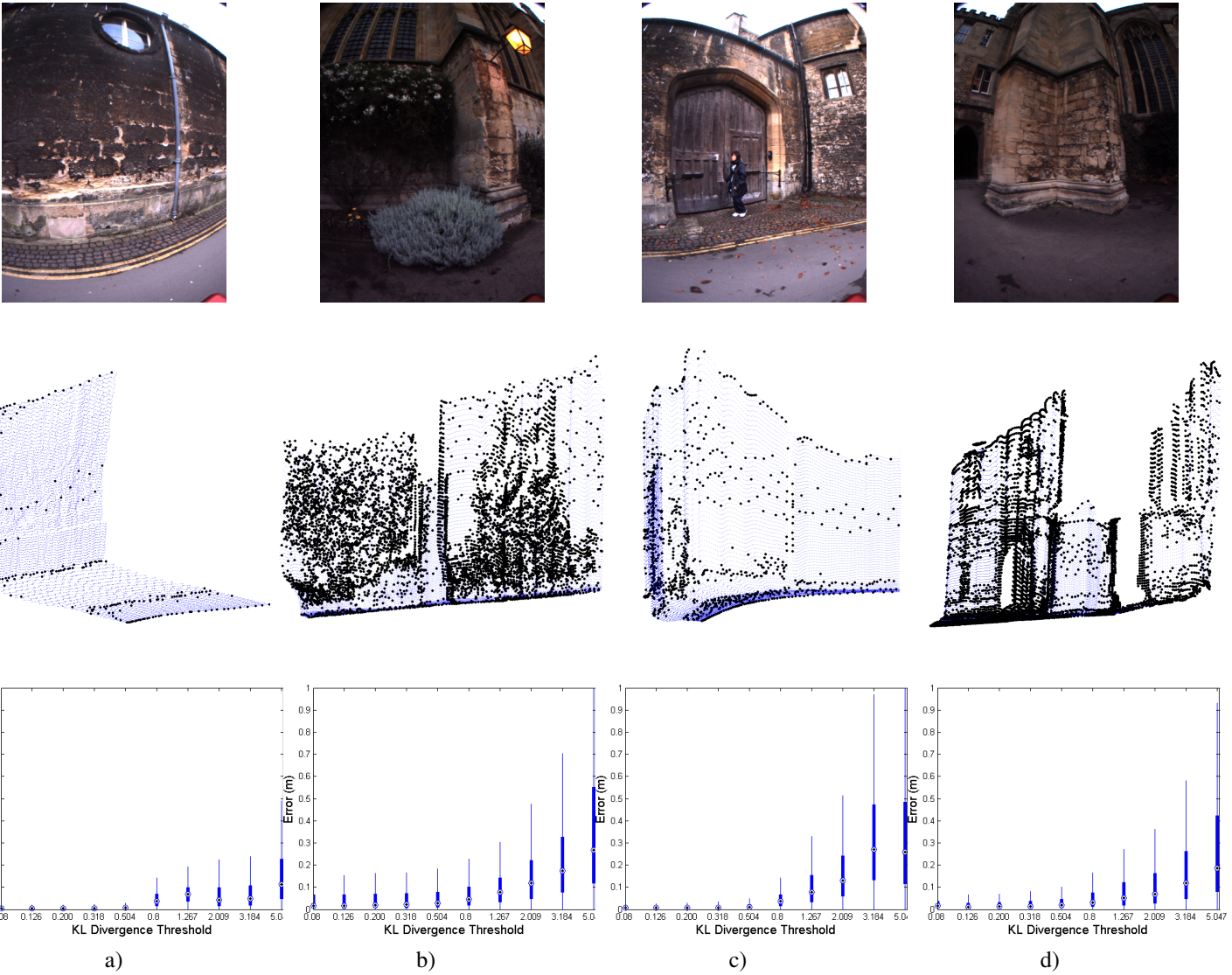
Figure 8. real-world examples (column-wise). Images (top) were captured at the same time as the laser data (2nd row) .Black points (2nd row) indicate actively sampled $\mathcal{D}'$ and blue, raw $\mathcal{D}$, in Euclidean space for $\kappa = 0.8\,nats$. One can clearly observe that intersections of planes, a person, bushes and other complicated regions of $\mathcal{W}$ are more heavily sampled than simple regions. Box Plots (bottom) denote median and interquartile range for each of the logarithmically spaced thresholds in Figure 7. Whiskers are plotted that extend a maximum of one and a half times the interquartile range.

Experimentation has provided insight into the compression ratio and surface reconstruction error that results from varying the KL divergence threshold used in our active sampling. To date we have considered the Matérn class of stationary covariance function, setting hyper parameters to best values found through search. We have experienced (Figure 10) that the accuracy of predictions in complex regions becomes that of the system with no subsampling. In order to enhance predictions in these complicated regions there is clear motivation to investigate possible alternative explicit basis functions such as chebychev polynomials, and non-stationary covariance functions such as [9], or [16]. The latter two approaches have the enticing benefit of automatically controlling hyper parameters, thereby allowing our predictions to be truly parameter free. We are also interested in online assessment of the active set size n to reduce computational load where a small n will suffice. In workspaces where the same regions are repeatedly mapped, one can imagine enormous compression ratios if we re-use the set $\mathcal{D}'$ that was created on the first iteration. In order to do this one would require a method to map between different sections of our beam space parameterisation. These developments are our next focus.

VII. Acknowledgments/Thanks

References

[1] J. Bares, M Hebert, T. Kanade, E. Krotkov, T. Mitchell, R. Simmons, and W. R. L. Whittaker. Ambler: An autonomous rover for planetary exploration. *IEEE Computer*, 22(1):18–26, 1989.
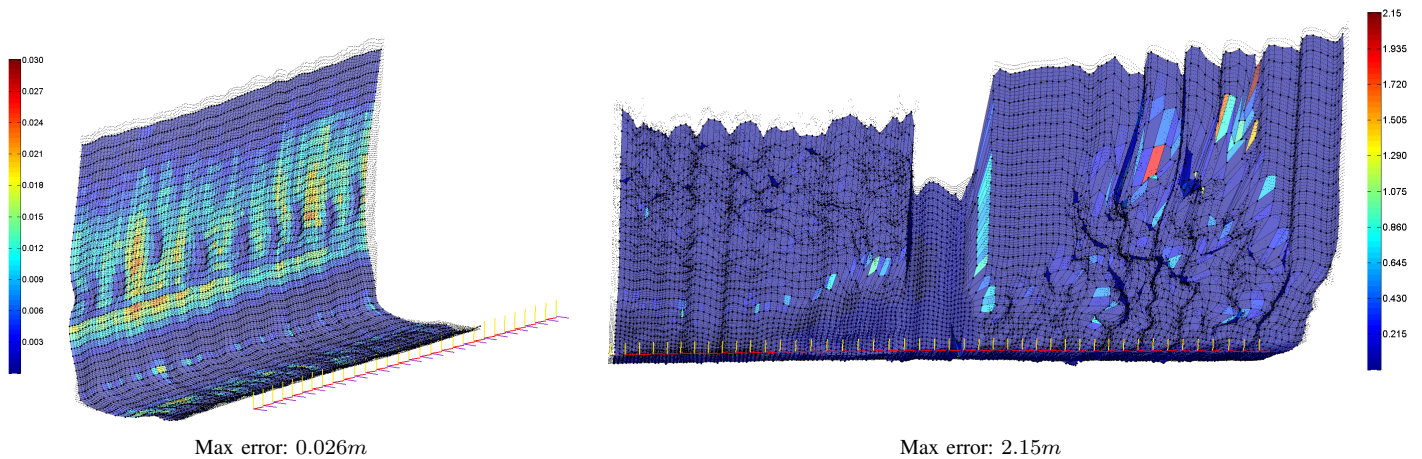
Max error: $0.026m$ Max error: $2.15m$

Figure 10. Surface predictions conditioned on $\mathcal{D}'$. Colouring is provided by the ℓ_1 norm in range from original points (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 environments of examples a) and b) from Figure 8 respectively. A profile view of a short section of (Left) is also plotted in Figure 9. In (left), where there is an absence of large discontinuities we notice the largest errors are away from $\mathcal{D}'$ (dark blue areas). In (right) largest errors occur 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 IV-B.

[2] L. Csató and M. Opper. Sparse on-line Gaussian processes. *Neural Computation*, 14(3):641–668, 2002.

[3] M.P. Deisenroth, C.E. Rasmussen, and J. Peters. Gaussian process dynamic programming. *Neurocomputing*, 72(7-9):1508–1524, 2009.

[4] J.J. Dongara, C.B. Moler, J.R. Bunch, and G.W. Stewart. *LINPACK users' Guide*. SIAM, 1979.

[5] C. Früh and A. Zakhor. Constructing 3d city models by merging aerial and ground views. *IEEE Comput. Graph. Appl.*, 23(6):52–61, 2003.

[6] D. Hähnel, W. Burgard, and S. Thrun. Learning compact 3D models of indoor and outdoor environment with a mobile robot. *Robotics and Autonomous Systems*, 44(1):15–27, 2003.

[7] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 19–26. ACM New York, NY, USA, 1993.

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

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

[10] M. C. Martin and H. Moravec. Robot evidence grids. Technical Report CMU-RI-TR-96-06, Robotics Institute, 1996.

[11] G. Matheron. The intrinsic random functions and their applications. *Advances in Applied Probability*, 5(3):439–468, 1973.

[12] The Mathworks. MATLAB 2008a. Natick, MA, 2008.

[13] P.M. Newman, G. Sibley, M. Smith, M. Cummins, A.R. Harrison, C. Mei, I. Posner, R. Shade, D. Schroeter, D.M. Cole, and I. Reid. Navigating, recognising and describing urban spaces with vision and laser. *IJRR*, 28(11-12):1406–1433, November 2009.

[14] D. Nguyen-Tuong and J. Peters. Using Model Knowledge for Learning Inverse Dynamics. In *IEEE International Conference on Robotics and Automation*, 2010.

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

[16] M.A. Osborne, S.J. Roberts, A. Rogers, S.D. Ramchurn, and NR Jennings. Towards real-time information processing of sensor network data using computationally efficient multi-output Gaussian processes. In *Proceedings of the 7th international conference on Information processing in sensor networks*, pages 109–120. IEEE Computer Society Washington, DC, USA, 2008.

[17] P. Pfaff and W. Burgard. An efficient extension of elevation maps for outdoor terrain mapping. In *Intl, Conf. on Field and Service Robotics (FSR)*, 2005.

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

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

[20] J. Quiñonero-Candela and C.E. Rasmussen. A unifying view of sparse approximate gaussian process regression. *J. Mach. Learn. Res.*, 6:1939–1959, 2005.

[21] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[22] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proc. 3DIM*, pages 145–152, 2001.

[23] B. Scholkopf and A.J. Smola. *Learning with kernels*. the MIT Press, 2002.

[24] M. Seeger. Low rank updates for the Cholesky decomposition. Technical report, University of California at Berkeley, 2008.

[25] M. Seeger, C.K.I. Williams, and N.D. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In *Workshop on AI and Statistics*, volume 9, 2003.

[26] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P.M. Newman. The new college vision and laser data set. *International Journal for Robotics Research (IJRR)*, 28(5):595–599, May 2009.

[27] M. L. Stein. *Interpolating Spatial Data*. Springer-Verlag, 1999.

[28] R. Triebel, W. Burgard, and F. Dellaert. Using hierarchical em to extract planes from 3d range scans. In *Int. Conf. Robotics and Automation (ICRA)*, 2005.

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

[30] S. Vasudevan, F. Ramos, E. Nettleton, H. Durrant-Whyte, and A. Blair. Gaussian process modeling of large scale terrain. In *Int. Conf. Robotics and Automation (ICRA)*, 2009.