Non-stationary dependent Gaussian processes for data fusion in large-scale terrain modeling

Shrihari Vasudevan, Fabio Ramos, Eric Nettleton and Hugh Durrant-Whyte
Australian Centre for Field Robotics, University of Sydney, NSW 2006, Australia
Email: shrihari.vasudevan@ieee.org, {f.ramos,e.nettleton,hugh}@acfr.usyd.edu.au

***Abstract*— Obtaining a comprehensive model of large and complex terrain typically entails the use of both multiple sensory modalities and multiple data sets. This paper demonstrates the use of dependent Gaussian processes for data fusion in the context of large scale terrain modeling. Specifically, this paper derives and demonstrates the use of a non-stationary kernel (Neural Network) in this context. Experiments performed on multiple large scale (spanning about 5 sq km) 3D terrain data sets obtained from multiple sensory modalities (GPS surveys and laser scans) demonstrate the approach to data fusion and provide a preliminary demonstration of the superior modeling capability of Gaussian processes based on this kernel.**

I. Introduction

Most field robotics applications such as mining and agriculture automation require robots to function in large and complex terrain. For autonomous robots to function in such high-value applications, an efficient, flexible and high-fidelity representation of space is critical. The key challenges in realizing this are that of dealing with the problems of uncertainty, incompleteness and handling highly unstructured terrain. Uncertainty and incompleteness are virtually ubiquitous in robotics as sensor capabilities are limited. The problem is magnified in a field robotics scenario due to sheer scale of the application (for instance, a mining or space exploration scenario). Contemporary tessellation based surface mapping approaches have not been able to provide a statistically sound solution to the problem of uncertainty incorporation and management. The assumption of statistical independence of data has resulted in many popular interpolation techniques being inaccurate in the context of modeling terrain.

Typically, sensory data is incomplete due to the presence of entities that occlude the sensors view. This is compounded by the fact that every sensor has limited perceptual capabilities i.e. limited range and applicability. Thus, most large scale modeling experiments would ideally require multiple sensory snapshots and multiple sensors to obtain a more complete model. These sensors may have different characteristics (e.g. range, resolution, accuracy). The problem thus is in fusing these multiple and multi-modal sensory data sets to obtain an integrated model - this is the theme of the paper. Terrain data can be obtained using numerous sensors including 3D laser scanners and GPS. The former provide dense and accurate data whereas a GPS based survey typically comprises of a relatively sparse set of well chosen points of interest.

This paper uses a Gaussian process (GP) representation of terrain data, as presented in [1]. The contribution of this work is a novel approach to fusing multiple, multi-modal terrain data sets to obtain a comprehensive model of the terrain under consideration. The fusion technique is generic and applicable as a general Gaussian process fusion methodology. The specific contribution of this work is the derivation and use of non-stationary kernels for multi-task problems with dependent processes. Experiments conducted using large scale 3D data obtained from GPS and laser scanner based surveys in real application scenarios (mining) are reported in support of the proposed approach.

II. Related work

State-of-the-art representations used in applications such as mining, space exploration and other field robotics scenarios as well as in geospatial engineering are typically limited to elevation maps ([2] and [3]), triangulated irregular networks (TIN's) ([4] and [5]), contour models and their variants or combinations ([6] and [7]). Each of these methods have their own strengths and preferred application domains. The former two are more popular in robotics. All of these representations, in their native form, do not handle spatially correlated data effectively and do not have a statistically principled way of incorporating and managing uncertainty.

Gaussian processes [8] (GP's) are powerful non-parametric Bayesian learning techniques that can handle these issues. Recently, Gaussian processes have been applied in the context of terrain modeling - see [9] and [1]. They produce a scalable multi-resolution model of the large scale terrain under consideration. They yield a continuous domain representation of the terrain data and hence can be sampled at any desired resolution. They incorporate and handle uncertainty in a statistically sound manner and represent spatially correlated data appropriately. They model and use the spatial correlation of the given data to estimate the elevation values for other unknown points of interest. In an estimation sense, GP's provide the best linear unbiased estimate [10] based on the underlying stochastic model of the spatial correlation between the data points. They basically perform an interpolation methodology called *Kriging* [11] which is a standard interpolation technique used in the mining industry. GP's thus handle both uncertainty and incompleteness effectively.

The work [1], also proposed the use of non-stationary kernels (neural network) to model large scale discontinuous spatial data. It compared performances of GP's based on stationary (squared exponential) and non-stationary (neural network) kernels as well as several other standard interpolation methods applicable to elevation maps and TIN's, in the context of large scale terrain modeling. The non-stationary neural network kernel was found to be superior to the stationary squared exponential kernel and at least as good as most standard interpolation techniques for a range

of terrain (in terms of sparsity/complexity/discontinuities). The work presented in this paper builds on GP terrain representation discussed. However it addresses the problem of fusing multiple such terrain representations into an integrated representation and focuses particularly on the neural-network kernel based GP's.

Data fusion in the context of Gaussian processes is necessitated by the presence of multiple, multi-modal, incomplete and uncertain data sets of the entity being modeled. Two preliminary attempts towards addressing this problem include [12] and [13]. The former bears a "hierarchical learning" flavor to it in that it demonstrates how a GP can be used to model an expensive process by (a) modeling a GP on an approximate or cheap process and (b) using the many input-output data from the approximate process and the few samples available of the expensive process together in order to learn a GP for the latter. The latter work attempts to generalize arbitrary transformations on GP priors through linear transformations. It hints at how this framework could be used to introduce heteroscedasticity (random variables with non-constant variance) and how information from different sources could be fused. However, specifics on how the fusion can actually be performed are beyond the scope of the work.

Two recent works that demonstrate data fusion (based on GP's) in the context of large scale terrain modeling include [14] and [15]. The former is based on the ideas that (a) data from the same entity can be modeled using a single set of GP hyperparameters with just the noise parameter varying between data sets i.e. the data sets are considered as different noisy samples of a common terrain that has to be modeled and (b) the fusion problem can then be treated as a standard GP regression/estimation problem with data having different noise parameters. The work [15] treats the data fusion problem as one of (a) modeling each data set using a GP and (b) formulating the data fusion problem as a conditional estimation problem wherein estimation of a GP is improved using information from other GP's - through learning auto-covariances and cross-covariances between them. This idea has been inspired by recent machine learning contributions in GP modeling ([16] and [17]), the latter approach being based on [18]. In kriging terminology, this idea is akin to co-kriging ([19]).

The work presented in this paper is a theoretical and practical extension to that presented in [15]. The state-of-the-art in multi-task/dependent GP modeling uses stationary kernels as computing closed form auto/cross-covariance functions for the selected kernel is a major complexity in applying this technique. This work extends the state-of-the-art by demonstrating the use of a non-stationary kernel (the neural network kernel) in the context of multi-task modeling using dependent Gaussian processes. Experiments are performed on large scale terrain data obtained from real mining scenarios. The scale of the experiments represents a distinctive feature of this work. Towards ensuring the scalability of the approach, approximation methods have been used in both the learning and inference stages. The contribution of this work is thus a novel method of fusing multiple multi-modal large scale data sets (terrain data, in this case) into an integrated model using non-stationary (neural-network) dependent GP's. Note that this work develops the fusion methodology. The registration of individual data sets to a common reference frame is assumed given for this work.

III. Approach

A. Gaussian processes

Gaussian processes ([8]) (GP's) are stochastic processes wherein any finite subset of random variables is jointly Gaussian distributed. They are non-parametric Bayesian, continuous representations that provide a powerful basis for modeling spatially correlated and possibly uncertain data. They may be thought of as a Gaussian probability distribution in function space. They are characterized by a mean function $m(\mathbf{x})$ and the covariance function $k(\mathbf{x}, \mathbf{x}')$ that together specify a distribution over functions. In the context of the problem at hand, each $\mathbf{x} \equiv (x, y)$ (2D coordinates) and $f(\mathbf{x}) \equiv z$ (elevation) of the given data. Although not necessary, the mean function $m(\mathbf{x})$ may be assumed to be zero by scaling the data appropriately such that it has an empirical mean of zero. The covariance function or kernel models the relationship between the random variables corresponding to the given data. The non-stationary neural network (NN) kernel ([20], [21] and [22]) takes the form

$$k_{NN}(\mathbf{x}, \mathbf{x}', \Sigma) = \frac{2}{\pi} \arcsin\left(\frac{2\tilde{\mathbf{x}}^T \Sigma \tilde{\mathbf{x}}'}{\sqrt{(1+2\tilde{\mathbf{x}}^T \Sigma \tilde{\mathbf{x}})(1+2\tilde{\mathbf{x}}'^T \Sigma \tilde{\mathbf{x}}')}}\right) \quad (1)$$

where $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{x}}'$ are augmented input vectors (each point is augmented with a 1), $\Sigma = \begin{bmatrix} \beta & 0 & 0 \\ 0 & l_x & 0 \\ 0 & 0 & l_y \end{bmatrix}^{-2}$ is the length-scale matrix, a measure of how quickly the modeled function changes in the directions x and y with β being a bias factor and d being the dimensionality of the data. The variables l_x , l_y , β constitute the kernel hyperparameters. The NN kernel represents the covariance function of a neural network with a single hidden layer between the input and output, infinitely many hidden nodes and using a Sigmoid as the transfer function [21] for the hidden nodes. Hornik in [23] showed that such neural networks are universal approximators and Neal [20] observed that the functions produced by such a network would tend to a Gaussian process.

Regression using GP's uses the fact that any finite set of training (evaluation) data and test data of a GP are jointly Gaussian distributed. This idea, shown in Equation 2, yields the standard GP regression equations 3 and 4 which respectively represent the mean-value and the uncertainty in the prediction.

$$\begin{bmatrix} \mathbf{z} \\ f_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right) \quad (2)$$

$$\bar{f}_* = K(X_*, X)[K(X,X) + \sigma_n^2 I]^{-1}\mathbf{z} \quad (3)$$

$$\mathrm{cov}(f_*) = K(X_*, X_*) - K(X_*, X)[K(X,X) + \sigma_n^2 I]^{-1} K(X, X_*) \quad (4)$$

For n training points and n_* test points, $K(X, X_*)$ denotes the $n \times n_*$ matrix of covariances evaluated at all pairs of training and test points. The terms $K(X, X)$, $K(X_*, X_*)$ and $K(X_*, X)$ can be defined likewise. The function values (f_*) corresponding to the test locations (X_*) given the training inputs X (a set of $\mathbf{x}$), training outputs $\mathbf{z}$ (a set of elevation values $f(\mathbf{x})$) and the kernel are given by Equation 3 and their uncertainties, by Equation 4.

B. Multi-output / Dependent Gaussian processes

Multi-output Gaussian processes (MOGP's or multi-task GP's) extend GP's to handle multiple correlated outputs simultaneously. The main advantage of this technique is that the model exploits not only the spatial correlation of data corresponding to one output but also those of the other outputs. This improves GP regression/prediction.

The objective is to model terrain data obtained as (x, y, z) coordinates from multiple and multi-modal datasets. An elevation map, at any chosen resolution and region of the terrain, needs to be estimated using these different datasets. This can be achieved by performing a conditional estimation given the different datasets and their GP models. The problem can thus be specified as estimating

$$\mathbb{E}[\, f_*(\mathbf{X}_*)\,] \;,\; var(f_*(\mathbf{X}_*)) \mid X_i \;,\; \mathbf{z_i} \;,\; GP_i \;,\; X_* \;, \quad (5)$$

where $X_i = (x_i, y_i)$ and $\mathbf{z_i} = z_i$ are the given data sets, GP_i are their respective GP model hyperparameters and i varies from 1 to the number of data sets available, henceforth denoted by nt. This estimation will need to take into account both the spatial correlation within each dataset as well as the spatial correlation across different datasets. Correlations between GP's can be modeled using auto-covariances and cross-covariances between them.

The process convolution approach ([18]) is a generic methodology which formulates a GP as a white noise source convolved with a smoothing kernel. Modeling the GP then amounts to modeling the hyperparameters of the smoothing kernel. The advantage of formulating GP's this way is that it readily allows the GP to be extended to model more complex scenarios, one such scenario being the multi-output or dependent GP's (MOGP's or DGP's). The following formulation for DGP's based on the NN kernel is inspired by [18] and [17]. A derivation is provided in the appendix.

Given that a single terrain is being modeled, a single Gaussian white noise process (denoted by $X(s)$ and representing (x, y) information of the data sets) is chosen as the underlying latent process. This process, when convolved with different smoothing kernel (denoted by k_i) produce different data sets. The smoothing kernel for the NN kernel takes the form

$$k(x, u) \;=\; \frac{1}{(2\pi)^{\frac{d+1}{4}} |\Sigma|^{\frac{1}{4}}} \;\mathrm{erf}(u^T \tilde{x}) \;\exp(\frac{-u^T \Sigma^{-1} u}{4}) \,. \quad (6)$$

The result of this convolution is denoted by $U_i(s)$. The observed data is assumed to be noisy and thus an additive white Gaussian noise $N(0, \sigma_i^2)$ (denoted by $W_i(s)$) is added to each process convolution output to yield the final data sets observed. Equations 7 and 8 show the mathematical formulation of the process convolution approach.

$$Y_i(s) \;=\; U_i(s) \;+\; W_i(s) \quad (7)$$

$$U_i(s) \;=\; \int_s k_i(s, \lambda) \;\star\; X(\lambda)\; d\lambda \quad (8)$$

Fusion GP regression takes into account data from the individual data sets as well as the auto and cross covariances between the respective GP's that model them. The auto-covariances and cross-covariances can be computed through a convolution integral as the kernel correlation, as demonstrated in [17]. Boyle et al. apply this technique for stationary squared exponential kernel. This work inspires from [18] and [17] to derive the auto and cross covariance functions for the non-stationary NN kernel. For two GP's $N(0, k_i)$ and $N(0, k_j)$ based on the NN kernel and with length scale matrices Σ_i and Σ_j respectively, the auto and cross-covariances are specified by Equation 9.

$$\begin{aligned} &K_{ij}^U(x, x') \;= \\ &K_f \,.\, 2^{\frac{1}{2}} \;|\Sigma_i|^{\frac{1}{4}} |\Sigma_i \;+\; \Sigma_j|^{-\frac{1}{2}} |\Sigma_j|^{\frac{1}{4}} \;k_{NN}(\mathbf{x}, \mathbf{x}', \Sigma_{ij}) \end{aligned} \quad (9)$$

where Σ_{ij} is obtained as $\Sigma_{ij} \;=\; 2\; \Sigma_i \;(\Sigma_i \;+\; \Sigma_j)^{-1}\; \Sigma_j$, The term, $k(\mathbf{x}, \mathbf{x}', \Sigma_{ij})$, is the NN kernel for two data $\mathbf{x}$, $\mathbf{x}'$ and length scale matrix Σ_{ij}. It is given by Equation 1. $K_{ii}^U \;(i \;=\; j)$ represents the auto-covariance of the i^{th} data set with itself and $K_{ij}^U \;(i \;\neq\; j)$ represents the cross covariance between the i^{th} and j^{th} data sets. These model the covariance between the input points of the data sets (x, y) and not the noisy observations. Thus, they do not consider the noise component of the observed data points (the elevation or z values). The K_f term in Equation 9 is inspired from [16]. This term models the task similarity between individual tasks (or data sets if only one task is being modeled). Incorporating it in the auto and cross covariances provides additional flexibility to the dependent GP modeling process. It is a symmetric matrix of size $nt * nt$ and is learnt along with the other GP hyperparameters.

The covariance matrix term $K(X, X)$ in Equations 3 and 4 is then specified as

$$K(X, X) \;=\; \begin{bmatrix} K_{11}^Y & K_{12}^Y & \dots & K_{1\,nt}^Y \\ K_{21}^Y & \dots & \dots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ K_{nt\,1}^Y & \dots & \dots & K_{nt\,nt}^Y \end{bmatrix} , \quad (10)$$

where

$$K_{ii}^Y \;=\; K_{ii}^U(X, X) \;+\; \sigma_i^2 I \quad (11)$$

$$K_{ij}^Y \;=\; K_{ij}^U(X, X) \quad (12)$$

K_{ii}^Y represents the auto-covariance of the i^{th} data set with itself and K_{ij}^Y represents the cross covariance between the i^{th} and j^{th} data sets. These terms model the covariance between the noisy observed data points (elevation or z values). They also take the noise components of the individual data sets / GP's into consideration. $K(X_*, X)$ denotes the covariance between the test data points and the sets of input data (from the individual data sets) that are used for GP regression. It is given by

$$K(X_*, X) = [K^U_{i1}(X_*, X_1)\,,\, K^U_{i1}(X_*, X_2)\,,\, \ldots\, K^U_{i\,nt}(X_*, X_{nt})] \tag{13}$$

where i is the output to be predicted - it can vary from 1 to nt. $K(X_*, X_*)$ represents the a priori covariance of the test points (uncertainty of prediction) and is specified by

$$K(X_*, X_*) = K^U_{ii}(X_*, X_*) + \sigma_i^2 \tag{14}$$

The noise term is added assuming the test points are as noisy as the data points of the i^{th} GP. Finally, $\mathbf{z}$ represents the sets of z data corresponding to the training data taken from each of the data sets.

$$\mathbf{z} = [\mathbf{z_1}\,,\, \mathbf{z_2}\,,\, \ldots\,,\, \mathbf{z_{nt}}] \tag{15}$$

The hyperparameters of the system that need to be learnt include $nt * (nt + 1)/2$ task similarity values, $nt * 3$ length scale values of the individual NN kernels and nt noise values corresponding to the noise in the observed data sets.

C. GP Learning and scalability considerations

GP learning and inference are computationally expensive operations in that both require matrix inversion. This operation is of cubic complexity with respect to the number of points in consideration. Thus, GP learning and inference approximations, introduced in [15], are used in this work. Both use an efficient hierarchical representation of the data-sets (a KD-tree was used) and implement a moving-window/nearest-neighbor approximation. The GP inference approximation uses the nearest data points (from individual data sets) to the query point for regression. In the GP learning approximation, a small set of training points are identified through uniform sampling. The KD-tree is then used to also select points in each of their neighborhoods as training points. Thus, "patches" of data are selected for training. GP learning then proceeds by using the maximum marginal likelihood framework (maximizing Equation 16). To further ensure scalability, a block-learning procedure is adopted to learn the GP models. Instead of learning with all training points at once, blocks of points are used in a sequential marginal likelihood computation process within the optimization step. The block size is pre-defined and depends on the computational resources available.

$$\begin{aligned}\log\, p(\mathbf{z}|X, \theta) = \; & -\tfrac{1}{2}\mathbf{z}^T K(X, X)^{-1}\mathbf{z} \\ & -\tfrac{1}{2}\log|K(X, X)| - \tfrac{N}{2}\log(2\pi),\end{aligned} \tag{16}$$

where N is the total number of training points across the all data sets and the other terms are as defined before.

IV. Experiments

Experiments were conducted on simulated data, multiple large scale single sensor data (RIEGL laser scanner) as well as multiple multi-sensor data (RIEGL laser scanner and GPS data) taken from real mining scenarios. These experiments demonstrate the fundamental concept (MOGP/DGP), demonstrate simultaneous elevation and color modeling, data fusion using multiple uni/multi-modal data. Cases of overlapping and non-overlapping data were also addressed. The experiments can be found in full in [24]. This paper will only present the data fusion experiments so as to serve the two objectives of this paper - demonstrate nonstationary NN-DGP based data fusion and provide a preliminary demonstration of the superior modeling capabilities of this kernel (in the context of data fusion) over the stationary kernel used in [15]. A more comprehensive comparison including cross-validation experiments (as in [1]) is currently being performed. The mean squared error (MSE) between the prediction and the ground truth (from data set) is used as the performance metric. Data sets were split into three parts - training, test and evaluation. The first part was used for learning the GP model, the second part was only used for MSE computation and finally, the first and third parts together (essentially, all data not in the second part) were used to perform GP regression at the MSE test points as well as any other query points.

A. Multiple overlapping single sensor data

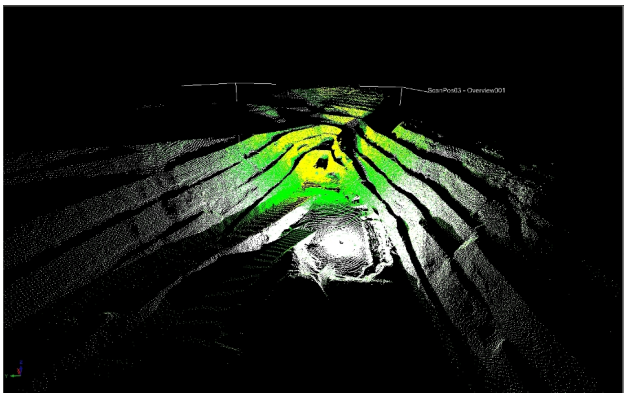

Fig. 1. West Angelas dataset of 3 overlapping RIEGL laser scans. Each scan had on an average about 500,000 points spread over about 1.8 x 0.5 sq km.

For this experiment, a large scale real world dataset comprising of 3 scans taken using a RIEGL LMSZ420 laser scanner at the West Angelas mine in Western Australia was used. Each scan had on an average about 500,000 points spread over about 1.8 x 0.5 sq km. The scans were overlapping to different extents. The objective of this experiment was to fuse the 3 scans to produce a more complete picture of the West Angelas mine. Figure 1 depicts the 3-scan dataset. Figure 2 shows the output obtained on applying the GP fusion methodology detailed in this report. The Mean Squared Error (MSE) was computed across a set of 10000 points, from the 1st data set, after each fusion step. These points may be selected uniformly, but patch-testing ([1] and [15]) is a more challenging and useful performance metric - hence 20 uniformly selected points along with 500 neighboring points each were used. Table I depicts the checks that were performed and corresponding results obtained. The uncertainty remains same or marginally less with each successive fusion step. Hence, the required condition for data fusion occurs. Further, it was observed that the MSE of the tested samples also decreased with each successive fusion step. This justified the use of data fusion in the context. The MSE values in parenthesis represent the values obtained for the same test conducted using a stationary Squared

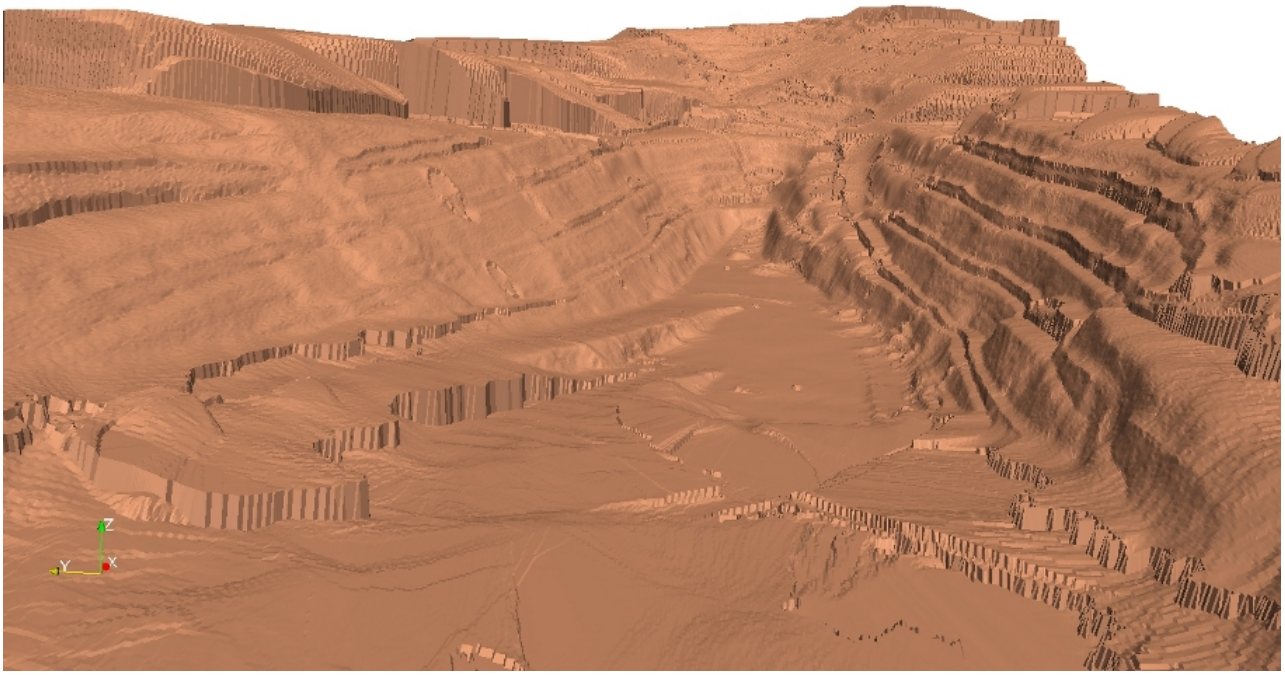

Fig. 2. Output of GP Fusion applied to the West Angelas dataset. The test data comprised of 1 Million points. The figure shows the surface map produced from the elevation output. Note the distinctive step like form on the side walls and the clearly visible roads into the pit.

Exponential kernel (SQEXP). Inline with the findings in [1], a non-stationary NN kernel based DGP produces superior performance compared to an SQEXP kernel based DGP.

TABLE I

GP FUSION USING NN-DGP: WEST ANGELAS DATA (3 SCANS, 500000 POINTS PER SCAN OVER 1.8 X 0.5 SQ KM, 5000 TRAINING POINTS PER SCAN, 10000 TEST POINTS - 20 PATCHES OF 500 POINTS EACH)

Scans	Mean Squared Error (MSE) (sq m)	Mean change in variance
Scan 1 only	0.4281 (0.8651)	
Scans 1 & 2	0.4265 (0.6145)	-3.38e-5 (no cases of increase in uncertainty)
Scans 1,2 & 3	0.4245 (0.5762)	-1.73e-5 (no cases of increase in uncertainty)

Note: MSE values obtained using a stationary SQEXP-DGP for the same test are provided in parenthesis for comparison.

B. Multiple multi-modal data sets

This experiment demonstrates data fusion of multiple multi-sensor data (RIEGL laser scanner and GPS survey) acquired from a large mine pit. Three data sets of the same area and of different characteristics were acquired from Mt. Tom Price mine in Western Australia. The first was a dense wide area (2146.6 m x 2302.1 m x 464.3 m) RIEGL laser scan comprising of over 850,000 points. The second was sparse GPS Survey having only about 34,530 points spread over 1437.2 m x 1879.5 m x 380.5 m. The third data set was a dense (about 400,000 points) RIEGL laser scan spread over a relatively smaller area as compared to the first scan (1416.6 m x 2003.4 m x 497.8 m). Figure 3 depicts the three data sets overlaid on each other to clarify the overall picture of the terrain in consideration.

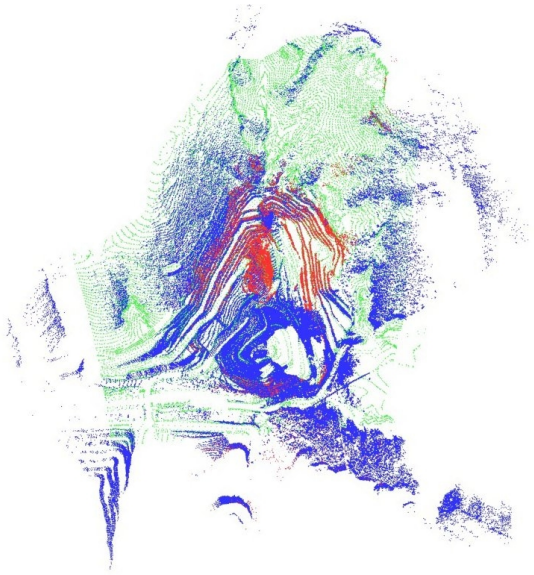

Fig. 3. The three Mt. Tom Price mine data sets (GPS survey and two laser scans) overlaid on one another for a clearer picture of the site in consideration. The points in blue represent Laser scan 1, the points in red represent the second laser scan and finally, the points in green represent the GPS data.

The objective was to demonstrate the benefits of GP data fusion using these data sets. The sparse GPS data is first modeled alone, then fused with the first laser data set and then the pair are fused with the third laser data set. The

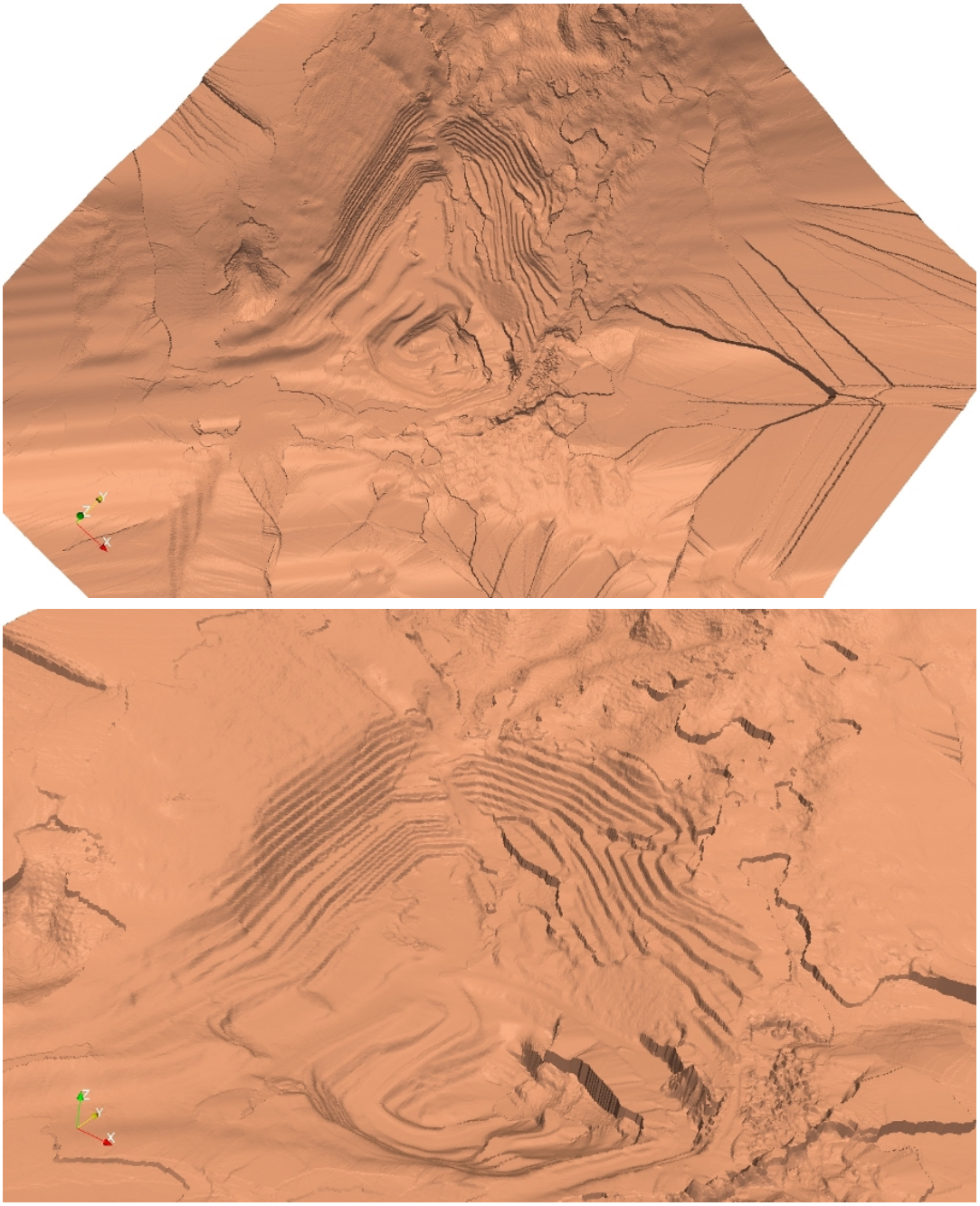

Fig. 4. Output of GP Fusion algorithm applied to the Tom Price data sets (GPS data and the two laser scanner data sets). The test data comprises of 1 million points. The surface map of the output elevation map is depicted in the image.

results of the fusion process are summarized in Table II. The results indicate the mean squared error (MSE) and average change of uncertainty for a set of test points (200 points with 50 neighbors each) from the first data set over successive steps of the fusion process. Figure 4 depicts the surface map obtained after fusing the GPS data with the two laser scanner data sets. As shown in Table II, the uncertainty decreases with each successive fusion step. Thus, the required condition for fusion occurs. Further, it is observed that the MSE also reduces with each fusion step. This justifies the benefits of data fusion in such a context. The MSE numbers in parenthesis represent the values obtained for the same test conducted using a stationary Squared Exponential kernel (SQEXP). Inline with findings in [1], an NN kernel based DGP produces superior performance compared to an SQEXP based DGP. Note that (1) larger test patches were used for the West Angelas mine data set as it is denser/flatter than the GPS data of the Mt. Tom Price data set (2) the reduction in MSE is less pronounced in the NN-DGP than the SQEXP-DGP; this is attributed to the superior modeling capability of the NN kernel based GP of the first data set and (3) the MSE values would be significantly reduced by performing the MSE computation over 10000 uniformly sampled points rather than patches (as each point would have supporting

data in its vicinity) and may also improve with further optimization.

TABLE II

GP FUSION USING NN-DGP: MT. TOM PRICE DATA - GPS DATA (2500 TRAINING POINTS), LASER SCAN 1 & 2 (5000 TRAINING POINTS) WITH 10000 TEST POINTS (200 PATCHES OF 50 POINTS EACH)

Fusion sequence (Data Sets)	Mean Squared Error (sqm)	Average change in variance
GPS data only	41.49 (99.84)	-
GPS data & Laser data 1	40.93 (93.33)	-0.0236 (no cases of increase in uncertainty)
GPS data, Laser data 1 & Laser data 2	40.28 (89.38)	-0.0096 (no cases of increase in uncertainty)

Note: MSE values obtained using a stationary SQEXP-DGP for the same test are provided in parenthesis for comparison.

V. CONCLUSION

This paper demonstrated the use of the multi-output/dependent Gaussian processes (MOGP's or DGP's) to fuse multiple multi-modal large scale terrain data sets. A key contribution of this work is the derivation and use of non-stationary (neural network) kernels in the context of modeling multi-task problems using dependent Gaussian processes. Real sensor data (GPS surveys and laser scans) taken from multiple mining scenarios were used to demonstrate the approach. The GP data fusion problem was cast as a conditional estimation using several *Dependent GP's*. The formalism could also be used to simultaneously model multiple aspects of the terrain as demonstrated in [15]. The proposed DGP based on the non-stationary (neural-network) kernel performed significantly better than the stationary squared exponential kernel based DGP ([15]) in fusing multiple terrain data sets. The scale of the experiments represents a distinctive feature of this work, enabled by the use of GP approximations in both learning and inference. The paper thus demonstrated a generic method of performing GP data fusion.

ACKNOWLEDGMENTS

This work has been supported by the Rio Tinto Centre for Mine Automation and the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government. The authors acknowledge the support of Annette Pal, Craig Denham, Joel Cockman and Paul Craine of Rio Tinto.

APPENDIX

DERIVATION OF THE CROSS-COVARIANCE FUNCTION FOR THE NEURAL NETWORK KERNEL

The process convolution approach [18] formulates a GP as a white noise source convolved with a smoothing kernel. Noisy observations are obtained by adding another white Gaussian noise $N(0, \sigma^2)$ to the convolution output. The work [17] was based on this work and applied it in the case of multi-output or dependent GP modeling using stationary squared exponential kernels. This derivation seeks to derive the auto and cross covariance functions for one non-stationary kernel - the neural network (NN) kernel. The following derivation is inspired from both [18] and [17]. The derivation is outlined below and given in more detail in the technical report version of this paper [24].

Given N outputs $Y_i(s)$ which are modeled using NN-GP's using smoothing kernel $k_i(s,\alpha)$ and are characterized by additive Gaussian white noise $W_i(s) = N(0, \sigma_i^2)$,

$$Y_i(s) = U_i(s) + W_i(s) \tag{17}$$

$$U_i(s) = \int_S k_i(s,\alpha)\, X(\alpha)\, d\alpha \tag{18}$$

so that, the covariance between two outputs $Y_i(s)$ and $Y_j(s)$ is given by

$$C_{ij}^Y(x_a, x_b) = C_{ij}^U(x_a, x_b) + \sigma_i^2 \delta_{ij} \delta_{ab} \tag{19}$$

$$\begin{aligned}
C_{ij}^U(x_a, x_b) &= E\left\{U_i(x_a)\, U_j(x_b)\right\} && (20)\\
&= E\left\{\int k_i(x_a,\alpha).X(\alpha)d\alpha \int k_j(x_b,\beta).X(\beta)d\beta\right\} \\
&= \int\int k_i(x_a,\alpha)\, k_j(x_b,\beta)\, E\left\{X(\alpha)\, X(\beta)\right\} d\alpha d\beta && (21)\\
&= \int\int k_i(x_a,\alpha)\, k_j(x_b,\beta)\, \delta(\alpha-\beta)\, d\alpha d\beta && (22)\\
&= \int k_i(x_a,\alpha)\, k_j(x_b,\alpha)\, d\alpha && (23)
\end{aligned}$$

The order of the integration and expectation is changed in Equation 21 because $\int |\, k_i(x_a,\alpha)\,|^2\, d\alpha < \infty$ for all i subject to the condition that the NN kernel is applied in a bounded neighborhood of data. Thus, the various $k_i(x_a,\alpha)$ are finite energy kernels and corresponding to stable linear filters so long as they are applied locally. $X(\alpha)$ and $X(\beta)$ are Gaussian white noise processes which will covary only when $\alpha = \beta$ and hence Equations 22 and 23.
The NN kernel is given by

$$k_{NN}(x, x', \Sigma) = \frac{1}{(2\pi)^{\frac{d+1}{2}} |\Sigma|^{\frac{1}{2}}} \int erf(\alpha^T \tilde{x})\, erf(\alpha^T \tilde{x'}) \exp(-\tfrac{1}{2}\alpha^T \Sigma^{-1} \alpha)\, d\alpha \tag{24}$$

This can be evaluated analytically (see appendix of [21]) to give

$$k_{NN}(x, x', \Sigma) = \frac{2}{\pi} \arcsin\left(\frac{2\tilde{x}\Sigma\tilde{x'}}{\sqrt{(1+2\tilde{x}^T\Sigma\tilde{x})(1+2\tilde{x'}^T\Sigma\tilde{x'})}}\right) \tag{25}$$

Let the smoothing kernel for the NN kernel be defined as

$$k(x,\alpha) = \frac{1}{(2\pi)^{\frac{d+1}{4}} |\Sigma|^{\frac{1}{4}}} erf(\alpha^T \tilde{x}) \exp(-\frac{1}{4}\alpha^T \Sigma^{-1} \alpha) \tag{26}$$

This is a non-stationary smoothing kernel as it relies on the dot product of x and α.Given a latent process (a Gaussian white noise process) $X(s)$, N-outputs $U_1(s) \ldots U_N(s)$ and N smoothing kernels $k_i(s)$, the auto-covariance and cross-covariance functions between the i^{th} and j^{th} outputs is given by Equation 23 as

$$C_{ij}^U(x, x', \Sigma_i, \Sigma_j) = \int_S k_i(x,\alpha)\, k_j(x',\alpha)\, d\alpha \tag{27}$$

S represents the domain of the data. For instance, $S \epsilon R^p$, p-dimensional real data. Using Equations 26 and 27, the auto-covariance and cross-covariance functions two NN-GP's can be derived (through simple algebraic manipulation) as

$$C_{ii}^U = k_{NN}(x, x', \Sigma_i) \tag{28}$$

$$C_{ij}^U = 2^{\frac{1}{2}} |\Sigma_i|^{\frac{1}{4}} |\Sigma_i + \Sigma_j|^{-\frac{1}{2}} |\Sigma_j|^{\frac{1}{4}}\, k_{NN}(x, x', \Sigma_{ij}) \tag{29}$$

where

$$\Sigma_{ij} = 2\,\Sigma_i\,(\Sigma_i + \Sigma_j)^{-1}\,\Sigma_j$$

and

$$k_{NN}(x, x', \Sigma) = \frac{2}{\pi} \arcsin\left(\frac{2\tilde{x}\Sigma\tilde{x'}}{\sqrt{(1+2\tilde{x}^T\Sigma\tilde{x})(1+2\tilde{x'}^T\Sigma\tilde{x'})}}\right)$$

Proof for PSD property of auto/cross covariance function:
Given M data sets $X_1, X_2, ..., X_M$ (or tasks, if multiple outputs are modeled simultaneously), the covariance matrix of the observations, that is used for GP regression is given by

$$K^Y(X_1, \ldots, X_M) = \begin{bmatrix} C^Y_{11} & C^Y_{12} & \ldots & C^Y_{1M} \\ C^Y_{21} & \ldots & \ldots & C^Y_{2M} \\ \vdots & \vdots & \vdots & \vdots \\ C^Y_{M1} & \ldots & \ldots & C^Y_{MM} \end{bmatrix} \quad (30)$$

$$K^Y(X_1, \ldots, X_M) = \begin{bmatrix} C^U_{11} & C^U_{12} & \ldots & C^U_{1M} \\ C^U_{21} & \ldots & \ldots & C^U_{2M} \\ \vdots & \vdots & \vdots & \vdots \\ C^U_{M1} & \ldots & \ldots & C^U_{MM} \end{bmatrix} + \begin{bmatrix} \sigma_1^2 & 0 & \ldots & 0 \\ 0 & \sigma_2^2 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & \ldots & \sigma_M^2 \end{bmatrix} \quad (31)$$

In the RHS of Equation 31, the second matrix is PSD ie. $\geqslant 0$. The objective is to prove the LHS is PSD. Hence, the first matrix in the RHS needs to be shown to be PSD. The individual C^U_{ij} are given by Equation 26. Consider the expression

$$Q = (A_1, A_2, \ldots, A_M) \begin{bmatrix} C^U_{11} & C^U_{12} & \ldots & C^U_{1M} \\ C^U_{21} & \ldots & \ldots & C^U_{2M} \\ \vdots & \vdots & \vdots & \vdots \\ C^U_{M1} & \ldots & \ldots & C^U_{MM} \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_M \end{bmatrix}$$

where A_i are sets or arbitrary real numbers. This results in

$$Q = \sum_{i=1}^{M} \sum_{j=1}^{M} A_i C_{ij} A j^T$$

Assuming that the i^{th} data set has N_i data, the above expression becomes

$$Q = \sum_{i=1}^{M} \sum_{j=1}^{M} \sum_{p=1}^{N_i} \sum_{q=1}^{N_j} a_{pi} C_{ij}(x_{pi}, x_{qj}) a_{qj}$$

Substituting Equation 29 in above expression,

$$Q = \sum_{i=1}^{M} \sum_{j=1}^{M} \sum_{p=1}^{N_i} \sum_{q=1}^{N_j} a_{pi} a_{qj} 2^{\frac{1}{2}} |\Sigma_i|^{\frac{1}{4}} |\Sigma_i + \Sigma_j|^{-\frac{1}{2}} |\Sigma_j|^{\frac{1}{4}} k_{NN}(x, x', \Sigma_{ij})$$

This is of the form

$$Q = \sum_{i=1}^{M} \sum_{j=1}^{M} \sum_{p=1}^{N_i} \sum_{q=1}^{N_j} a'_{pi} a'_{qj} k_{NN}(x, x', \Sigma_{ij})$$

for some real $a'_{pi}$and a'_{qj}. Now, $Q \geqslant 0$ because $k(x, x', \Sigma_{ij})$ is the NN kernel / covariance function (between x and x', for some set of hyperparameters Σ_{ij}) and is by definition PSD. Hence the first matrix in Equation 31 is PSD and hence the covariance matrix produced by the auto/cross covariance function derived above is PSD. □

REFERENCES

[1] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Gaussian Process Modeling of Large Scale Terrain," *Journal of Field Robotics*, vol. 26(10), pp. 812–840, 2009.

[2] S. Lacroix, A. Mallet, D. Bonnafous, G. Bauzil, S. Fleury, M. Herrb, and R. Chatila, "Autonomous rover navigation on unknown terrains: Functions and Integration," *International Journal of Robotics Research (IJRR)*, vol. 21(10-11), pp. 917–942, 2002.

[3] R. Triebel, P. Pfaff, and W. Burgard, "Multi-Level Surface Maps for Outdoor Terrain Mapping and Loop Closing," in *International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, October 2006.

[4] J. Leal, S. Scheding, and G. Dissanayake, "3D Mapping: A Stochastic Approach," in *Australian Conference on Robotics and Automation*, November 2001.

[5] I. Rekleitis, J. Bedwani, D. Gingras, and E. Dupuis, "Experimental Results for Over-the-Horizon Planetary exploration using a LIDAR sensor," in *Eleventh International Symposium on Experimental Robotics*, July 2008.

[6] H. Durrant-Whyte, "A Critical Review of the State-of-the-Art in Autonomous Land Vehicle Systems and Technology," Sandia National Laboratories, USA, Tech. Rep. SAND2001-3685, November 2001.

[7] I. D. Moore, R. B. Grayson, and A. R. Ladson, "Digital terrain modelling: A review of hydrological, geomorphological, and biological applications," *Hydrological Processes*, vol. 5-1, pp. 3–30, 1991.

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

[9] C. Plagemann, S. Mischke, S. Prentice, K. Kersting, N. Roy, and W. Burgard, "A Bayesian regression approach to terrain mapping and an application to legged robot locomotion," *Journal of Field Robotics*, vol. 26(10), pp. 789–811, 2009.

[10] P. K. Kitanidis, *Introdcution to Geostatistics: Applications in Hydrogeology*. Cambridge University Press, 1997.

[11] G. Matheron, "Principles of Geostatistics," *Economic Geology*, vol. 58, pp. 1246–1266, 1963.

[12] M. El-Beltagy and W. Wright, "Gaussian processes for model fusion," in *International Conference on Artificial Neural Networks (ICANN)*, 2001.

[13] R. Murray-Smith and B. Pearlmutter, *Deterministic and Statistical Methods in Machine Learning, LNAI 3635*. Springer-Verlag, 2005, ch. Transformations of Gaussian Process priors, pp. 110–123.

[14] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Heteroscedastic Gaussian processes for data fusion in large scale terrain modeling," in *the International Conference for Robotics and Automation (ICRA)*, 2010.

[15] ——, "Large-scale terrain modeling from multiple sensors with dependent Gaussian processes," in *in the proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, October 2010.

[16] E. Bonilla, K. M. Chai, and C. Williams, "Multi-task Gaussian process prediction," in *Advances in Neural Information Processing Systems 20*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. Cambridge, MA: MIT Press, 2007, pp. 153–160.

[17] P. Boyle and M. Frean, "Dependent Gaussian processes," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2004, pp. 217–224.

[18] D. Higdon, *Quantitative Methods for Current Environmental Issues*. Springer, 2002, ch. Space and Space-Time Modeling Using Process Convolutions, pp. 37–54.

[19] H. Wackernagel, *Multivariate geostatistics: an introduction with applications*. Springer, 2003.

[20] R. M. Neal, *Bayesian Learning for Neural Networks*, ser. Lecture Notes in Statistics 118. New York: Springer, 1996.

[21] C. K. I. Williams, "Computation with infinite neural networks," *Neural Computation*, vol. 10(5), pp. 1203–1216, 1998.

[22] ——, "Prediction with Gaussian processes: From linear regression to linear prediction and beyond," in *Learning in Graphical Models*, M. I. Jordan, Ed. Springer, 1998, pp. 599–622.

[23] K. Hornik, "Some new results on neural network approximation," *Neural Networks*, vol. 6(8), pp. 1069–1072, 1993.

[24] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Non-stationary dependent Gaussian processes for data fusion in large scale terrain modeling," Australian Centre for Field Robotics, The University of Sydney, Tech. Rep., 2010.