Australian Centre for Field Robotics
A Key Centre of Teaching and Research

The Rose Street Building J04
The University of Sydney 2006 NSW Australia

Data fusion with Gaussian processes

Shrihari Vasudevan

T: + 61 2 9114 0891
E: shrihari.vasudevan@ieee.org

Technical Report ACFR-TR-2012-001

03 November 2011
Released 20 June 2012

Abstract

This report addresses the problem of fusing multiple sets of heterogeneous sensor data using Gaussian processes. Experimental studies from the context of large scale terrain modeling in a mining automation scenario are presented. Three techniques in increasing order of model complexity are discussed. The first is based on adding data to an existing Gaussian process model. The second approach treats data from different sources as different noisy samples of a common underlying terrain and fusion is performed using heteroscedastic Gaussian processes. The final approach models each data set by a separate GP model and learns spatial correlations between the data sets through auto and cross covariances. This approach to data fusion is based on dependent Gaussian processes. All three approaches are grounded on the basic Gaussian process model. The three approaches and multiple variants of them are evaluated. The contribution of this work is thus a unifying view of various approaches to data fusion using Gaussian processes, an evaluation that compares these approaches and multiple previously untested variants of them and an insight into the effect of model complexity on data fusion. The experiments prove that depending on the data set being modeled, simpler approaches can compete with or even outperform the most general approach to data fusion using Gaussian processes.

Acknowledgments

I am grateful to Lakshmi Vasudevan, Steve Scheding, Eric Nettleton, Fabio Ramos and Hugh Durrant-Whyte for their support. This work has been funded by the Rio Tinto Centre for Mine Automation.

Contents

1 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, sensor data is incomplete due to the presence of entities that occlude its field of 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 sensor data sets to obtain an integrated model. This theme of the report is studied in the context of large scale terrain modeling in a mining automation scenario.

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 report uses a Gaussian process (GP) representation of terrain data, as presented in [1]. Data fusion using Gaussian processes has been demonstrated using two methods in [2], [3] and [4]; the experimental results in these works being preliminary findings. The contribution of this work is the answering of two important questions - (1) how are the various GP data fusion approaches related and (2) which model should be used in a given context? Towards this contribution, this report reviews three basic approaches (the aforementioned two together with another fundamental approach) and unifies them within a common GP framework. It then evaluates the fundamental approaches and multiple previously unreported variants of them through statistically representative cross validation methods and draws inferences on the applicability of the approaches from the results obtained. The report is thus a comprehensive reference on the the state-of-the-art in this area. Experiments are performed on multiple large scale (spanning about 5 sq km) 3D terrain data sets obtained from multiple sensory modalities (GPS surveys and laser scans). The size of the data sets used in this work is a distinguishing aspect of this body of work. The fusion techniques discussed are generic and applicable as general Gaussian process fusion methodologies in any context.

2 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 ([5] and [6]), triangulated irregular networks (TIN's) ([7] and [8]), contour models and their variants or combinations ([9] and [10]). 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 [11] (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 [1] and [12]. 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 [13] based on the underlying stochastic model of the spatial correlation between the data points. They basically perform an interpolation methodology called *Kriging* [14] 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 report builds on this GP terrain representation. However, it addresses the problem of fusing multiple such terrain representations into an integrated representation.

Data fusion in the context of Gaussian processes is necessitated by the presence of multiple, multimodal, incomplete and uncertain data sets of the entity being modeled. Two preliminary attempts towards addressing this problem include [15] and [16]. 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. The work [17] integrates heterogeneous information within a Gaussian process classification setting, in a protein fold recognition application domain. Each feature representation is represented by a separate GP. The fusion uses the fact that individual feature representations are considered independent and hence a composite covariance function would be defined in terms of a linear sum of Gaussian process priors. This report addresses the data fusion problem wherein multiple, possibly heterogeneous sources of information are correlated. The objective is to use this correlation to improve the estimate of the quantity being modeled (terrain surface in this report). A recent work [18] integrates "hard" data obtained from sensors with "soft" information from human sources within a Gaussian process classification framework, by using a separate kernel for each data type and combining all the kernels using a product rule. This problem/approach is different from the work presented here. It uses heterogeneous information sources as mutually independent sources of information that are transformed into the kernel representation and combined using a product rule. The approaches presented in this report improve the estimate of the quantity (or quantities) being modeled by adequately/explicitly modeling and using the correlation between multiple heterogeneous information sources.

Two recent approaches demonstrating data fusion with Gaussian processes in the context of large scale terrain modeling were based on heteroscedastic GP's (HGP's) [2] and dependent GP's (DGP's) ([3] and [4]). The work [2] treated the data-fusion problem as one of combining different noisy samples of a common entity (terrain) being modeled. In the Machine Learning community, this idea is referred to as heteroscedastic GP's ([19], [20], [21] and [22]). The works [3] and [4] treated the data fusion problem as one of improving GP regression through modeling the spatial correlations (auto and cross covariances) between several dependent GP's representing the respective data sets. This idea has been inspired by recent machine learning contributions in GP modeling including [23] and [24], the latter being based on [25]. In kriging terminology, this idea is akin to co-kriging ([26]).

The contributions of this work over prior work (including those of the author [2], [3] and [4]) are -

- the unifying presentation of three approaches to data fusion using Gaussian processes. The aforementioned two approaches along with another fundamental approach based on Gaussian processes themselves, are presented. This analysis clarifies how the methods relate to each other.

- the statistically representative evaluation of the three approaches along with multiple previously unreported variants of them and a benchmark naive approach to data fusion. The experimental results in [2], [3] and [4] were preliminary in that they were neither statistically representative tests nor did they compare/benchmark their performance against other approaches.

- The analysis of the effect of model complexity and increased data availability on data fusion and the conclusions drawn thereof on the methodology of choosing a data fusion approach for a given context. The report proves that the most general approach is not always necessary or best for the task; a simpler approach may be just as effective or even better than the most general one, given the complexity of the data set being modeled. The choice of a suitable kernel is also critical to obtaining good results.

The fusion techniques discussed are generic and applicable as general Gaussian process fusion methodologies in any context. This work is thus meant to tie together past work in data fusion using Gaussian processes and be a single point reference to the state-of-the-art in this area.

3 Approach

3.1 Gaussian processes

Gaussian processes ([11]) (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/shifting 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. It can take numerous forms (see chapter 4 in [11]). The stationary squared exponential (or Gaussian) kernel (SQEXP) is given by

$$k_{SQEXP}(\mathbf{x}, \mathbf{x}', \Sigma) \;=\; \sigma_f^2 \, \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \Sigma (\mathbf{x} - \mathbf{x}')\right) , \tag{1}$$

where k is the covariance function or kernel; $\Sigma = \begin{bmatrix} l_x & 0 \\ 0 & l_y \end{bmatrix}^{-2}$ is a d x d length-scale matrix (d = dimensionality of input = 2 in this case), a measure of how quickly the modeled function changes in the directions x and y; σ_f^2 is the signal variance. The set of parameters l_x , l_y , σ_f are referred to as the kernel hyperparameters. The non-stationary neural network (NN) kernel ([27], [28] and [29]) takes the form

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

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 a $(d + 1)$ x $(d + 1)$ 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 input 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 Sigmoidal transfer function [28] for the hidden nodes. Hornik in [30] showed that such neural networks are universal approximators and Neal [27] observed that the functions produced by such a network would tend to a Gaussian process. Prior work ([1]) has found the NN kernel to be more effective than the SQEXP kernel at modeling discontinuous data.

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. Assuming noise free data, this idea is shown in Expression 3 (hereafter referred to as Equation 3). This leads to the standard GP regression equations yielding an estimate (the mean value, given by Equation 4) and its uncertainty (Equation 5).

$$\begin{bmatrix} \mathbf{z} \\ f_* \end{bmatrix} \;\sim\; N\left(0, \begin{bmatrix} K(X, X) & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right) \tag{3}$$

$$\bar{f}_* \;=\; K(X_*, X)\, K(X, X)^{-1}\, \mathbf{z} \tag{4}$$

$$\mathrm{cov}(f_*) \;=\; K(X_*, X_*) - K(X_*, X)\, K(X, X)^{-1}\, K(X, X_*) \tag{5}$$

For n training points $(X, \mathbf{z}) \;=\; (\mathbf{x_i}, z_i)_{i=1\ldots n}$ and n_* test points (X_*, f_*), $K(X, X_*)$ 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 $K(X_*, X)$ are defined likewise. In the event that the data being modeled is noisy, a noise hyperparameter (σ) is also learnt with the other GP hyperparameters and the covariance matrix of the training data $K(X, X)$ is replaced by $[K(X, X) + \sigma^2 I]$ in Equations 3, 4 and 5.

3.2 Data fusion using Gaussian processes

The general data fusion problem addressed in this report can be described as follows. Given multiple data sets (possibly multi-modal) of the terrain being modeled, the objective of the data fusion problem

is to estimate the elevation at a point given all prior data sets and the respective GP's (hyperparameters) that are used to model them. This can be specified as

$$\mathbb{E}[\, f_*(\mathbf{X}_*)\,]\ ,\ var(f_*(\mathbf{X}_*)) \mid \mathbf{X_i}\ ,\ \mathbf{z_i}\ ,\ GP_i\ ,\ \mathbf{X}_*\ , \tag{6}$$

where $(\mathbf{X_i}, \mathbf{z_i})$ are the given data sets, GP_i are their respective GP model hyperparameters, $\mathbf{X}_*$ is the set of test points to be evaluated ($= X_*$ if the points are from one data set) and i varies from 1 to the number of data sets available, denoted hereafter by nt.

The simplest method to accomplish data fusion using GP's is to use the additional data within the existing GP model. The assumption here is that the existing GP model (kernel and hyperparameters) provides an adequate representation for the new data. In this case, all of the GP_i above would be identical and only the data points, $\mathbf{X_i}$ and $\mathbf{z_i}$ need to be considered for the data fusion process. Equations 3, 4 and 5 respectively represent the data fusion model, regression estimates and uncertainties subject to the following modifications to the basic notation. The set

$$\mathbf{z} = [\ \mathbf{z}_1\ ,\ \mathbf{z}_2\ ,\ \mathbf{z}_3\ ,\ \ldots\ ,\ \mathbf{z}_{nt}\ ]'$$

represents the output elevation values of the selected training data from the individual data sets. The term

$$X = [\ X_1\ ,\ X_2\ ,\ X_3\ ,\ \ldots\ ,\ X_{nt}\ ]$$

denotes the input location values of the selected training data from the individual data sets. The covariance matrix of the training data is given by

$$K(X,X)\ \equiv\ \begin{bmatrix} K(X_1,X_1)+\sigma^2 I & K(X_1,X_2) & \ldots & K(X_1,X_{nt}) \\ K(X_2,X_1) & \ldots & \ldots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ K(X_{nt},X_1) & \ldots & \ldots & K(X_{nt},X_{nt})+\sigma^2 I \end{bmatrix}$$

and the covariance matrix between the test points and training points is given by

$$K(X_*,X)\ =\ [K(X_*,X_1), K(X_*,X_2), \ldots, K(X_*,X_{nt})]\ .$$

The matrix $K(X,X_*)$ is defined likewise. Finally, $K(X_*,X_*)$ represents the covariance matrix of the test points. If the test points are assumed to be as noisy as the training points, the covariance matrix of the test points will take the form

$$K(X_*,X_*)+\sigma^2 I\ .$$

3.3 Data fusion using heteroscedastic Gaussian processes

The heteroscedastic Gaussian process (HGP) data fusion methodology is based on two underlying ideas

1. Data from the same entity can be modeled using a single set of GP hyperparameters with just the noise parameter varying between data sets. Thus, the data sets are considered as different noisy samples of a common terrain that has to be modeled.

2. The fusion problem is treated as a standard GP regression/estimation problem with data having different noise parameters. The formulation is similar to the heteroscedastic GP formulation described in [19] and [22].

Consider the data fusion problem described in Equation 6. Equations 3, 4 and 5 respectively represent the HGP data fusion model, the regression estimates and their uncertainties, subject to the following modifications to the basic notation. The set

$$\mathbf{z} = [\ \mathbf{z}_1\ ,\ \mathbf{z}_2\ ,\ \mathbf{z}_3\ ,\ \ldots\ ,\ \mathbf{z}_{nt}\ ]'$$

represents the output elevation values of the selected training data from the individual data sets. The term

$$X = [\ X_1\ ,\ X_2\ ,\ X_3\ ,\ \ldots\ ,\ X_{nt}\ ]$$

denotes the input location values of the selected training data from the individual data sets. The covariance matrix of the training data is given by

$$K(X,X)\ \equiv\ \begin{bmatrix} K(X_1,X_1)+\sigma_1^2 I & K(X_1,X_2) & \ldots & K(X_1,X_{nt}) \\ K(X_2,X_1) & \ldots & \ldots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ K(X_{nt},X_1) & \ldots & \ldots & K(X_{nt},X_{nt})+\sigma_{nt}^2 I \end{bmatrix}\ .$$

Here, each data set i is associated with a noise parameter σ_i. These are factored into the covariance matrix of the training data. Any kernel ([11]) may be used so long as the same kernel is used for modeling each of the data sets. As in data fusion with simple GP's, the covariance matrix between the test points and training points is given by

$$K(X_*, X) \;=\; [K(X_*, X_1), K(X_*, X_2), \ldots, K(X_*, X_{nt})] \;.$$

The matrix $K(X, X_*)$ is defined likewise. Finally, the covariance of the test points is given by $K(X_*, X_*)$ assuming that they are noise free. If the predictions need to be made at test points that are as uncertain as the data at hand, the covariance of the test points may be specified by

$$K(X_*, X_*) + R(X_*)$$

so that Equation 5 will be modified to

$$\mathrm{cov}(f_*) = K(X_*, X_*) + R(X_*) \;-\; K(X_*, X)\, K(X, X)^{-1}\, K(X, X_*)\,. \tag{7}$$

Here, $R(X_*)$ represents the noise or uncertainty of the query points themselves. Typically these are not known. For heteroscedastic GP regression, estimation of the noise hyperparameters of the data points as well as the query points is a key issue. The works [19] and [22] deal with the problem by maintaining two GP's - one to estimate the quantity of interest given the *expected* values of the noise parameters (in addition to the data sets and GP hyperparameters) and the other GP to estimate the noise hyperparameters given the data points and query points. The former is a straightforward application of Equations 4 and 7. The latter GP is the key issue as it provides the noise values to the former GP. Both [19] and [22] make an intuitive approximation - the noise values obtained from the second GP are approximated by their expected values. This work adopts the same idea but implements it differently. As the query points can be assumed to be as noisy as the training data and the fact that this work adopts a local approximation methodology towards GP regression [1], the query points are assigned a noise value that is the expected value of the noise terms of data taken from the individual data sets. The individual noise terms are learnt as before. Thus,

$$R(x_*) \;=\; \frac{\Sigma_{i=1}^{n} N_i \sigma_i^2}{\Sigma_{i=1}^{n} N_i}\,,$$

where $x_* \,\epsilon\, X_*$ is a query point, N_i are the number of training data points chosen from the i^{th} data set and σ_i^2 is the noise variance of the GP modeling this data set. An inverse distance based weighted average of the noise values could also be used in this context.

Equations 4 and 5 (or 7) provide the batch fusion estimator, ie. they provide the conditional mean and covariance of the elevation given all the data sets taken together. As shown in [2], one could also cast this as a recursive fusion estimator where information is incrementally fused with existing data. It can be shown that the formalism guarantees that with the addition of data sets (any number, from any sensor), the uncertainty in the fused elevation estimate *cannot* increase. If the new or incoming data set has relevant information for the prediction at a query point in the first data set, the posterior uncertainty will decrease; if there is no relevant information (assume, for instance, no points are selected from successive data sets for a particular query point), the uncertainty will remain same. The data sets may thus be fused to generate integrated and comprehensive terrain models.

3.4 Data fusion using multi-output / dependent Gaussian processes (DGP's)

Multi-output Gaussian processes (MOGP's or multi-task GP's) or Dependent GP's (DGP's) extend Gaussian processes 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 of an output given the others, thus performing data fusion. A simple example of the concept is shown in Figure 1

As before, consider the data fusion problem described in Equation 6. Equations 3, 4 and 5 represent respectively the DGP data fusion model, the regression estimates and their uncertainties, subject to the following modifications to the basic notation. The set

$$\mathbf{z} = [\; \mathbf{z}_1 \;,\; \mathbf{z}_2 \;,\; \mathbf{z}_3 \;,\; \ldots \;,\; \mathbf{z}_{nt} \;]'$$

represents the output elevation values of the selected training data from the individual data sets. The term

$$X = [\; X_1 \;,\; X_2 \;,\; X_3 \;,\; \ldots \;,\; X_{nt} \;]$$

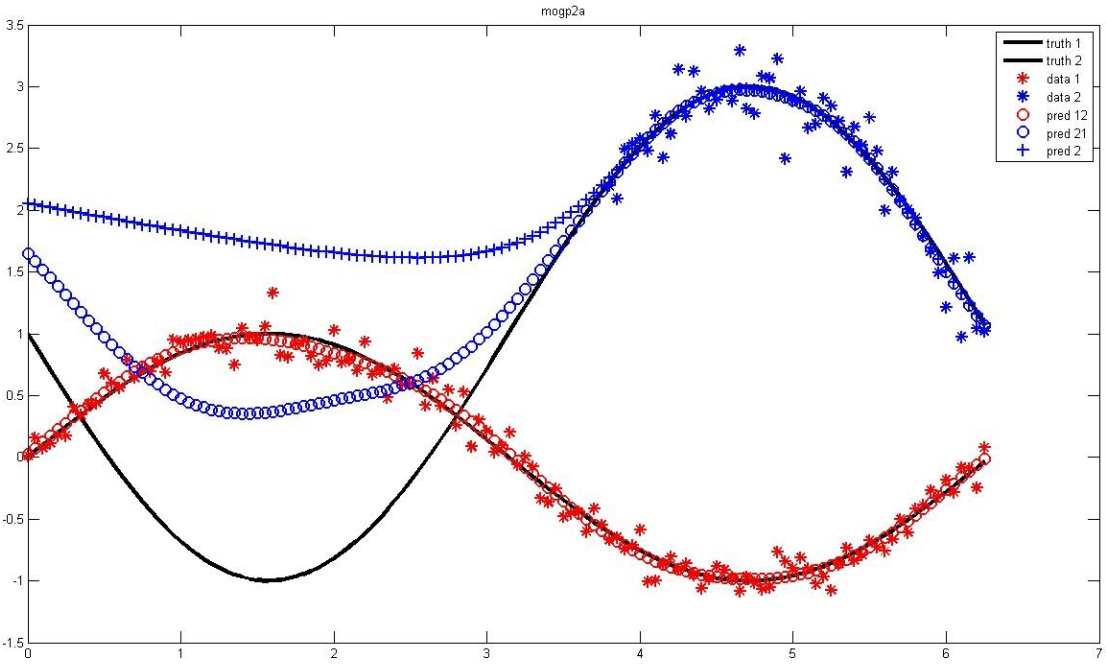

Figure 1: A simple demonstration of the DGP / MOGP concept. Two sine waves (black) are to be modeled. One is an inverted version of the other. Further noisy samples are available all over one of them (red) whereas the other one has noisy samples only in one part of it (blue). Merely using these few samples would result in a poor prediction of the sine wave in the areas devoid of samples. Using the spatial correlation with the red sampled sine wave enables the DGP/MOGP approach to improve the prediction of the blue sampled sine wave.

denotes the input location values of the selected training data from the individual data sets. Any kernel ([11]) may be used and even different kernel could be used for different data sets using the technique demonstrated in [31] (for stationary kernel) or the convolution process technique demonstrated in [25], [24] and in this report (for both stationary and nonstationary kernel). The covariance matrix of the training data is given by

$$K(X,X) \equiv \begin{bmatrix} K^Y_{11} & K^Y_{12} & \dots & K^Y_{1\,nt} \\ K^Y_{21} & \dots & \dots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ K^Y_{nt\,1} & \dots & \dots & K^Y_{nt\,nt} \end{bmatrix}$$

where

$$\begin{aligned} K^Y_{ii} &= K^U_{ii}(X_i,X_i) + \sigma_i^2 I \\ K^Y_{ij} &= K^U_{ij}(X_i,X_j)\,. \end{aligned}$$

Here, K^Y_{ii} represents the auto-covariance of the i^{th} data set with itself and K^Y_{ij} 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). Thus, they also take the noise components of the individual data sets / GP's into consideration. The corresponding noise free terms are respectively given by K^U_{ii} and K^U_{ij}. These are derived by using the process convolution approach to formulating Gaussian processes; details of this follow in the subsequent paragraphs. The covariance matrix between the test points and training points is given by

$$K(X_*,X) = [K^U_{i1}(X_*,X_1), K^U_{i2}(X_*,X_2), \dots, K^U_{i\,nt}(X_*,X_{nt})]\,,$$

where $i \,\epsilon\, \{1 \dots nt\}$ is the GP that is being evaluated given all other GP's. The matrix $K(X,X_*)$ is defined likewise. Finally, the covariance of the test points is given by

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

assuming the i^{th} GP needs to be evaluated for the particular test point. The mean and variance of the elevation estimate can thus be obtained by applying Equations 4 and 5, after incorporating multiple data

sets, multiple GP/noise hyperparameters and deriving appropriate auto and cross covariances functions that model the spatial correlation between the individual data sets. The data sets may thus be fused to generate integrated and comprehensive terrain models.

The process convolution approach ([25]) 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 was inspired by [25] and [24]. The work [3] was based on the SQEXP kernel [24]. In [4], the formulation was used to derive appropriate auto and cross covariance terms for the nonstationary NN kernel. The detailed derivation for the NN kernel is provided in the appendix of this report for completeness.

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 SQEXP covariance function used by [24] takes the form

$$k(x,u) \;=\; \exp\left(-\frac{1}{2}(x-u)^T\Sigma(x-u)\right). \tag{8}$$

The smoothing kernel for the NN covariance function takes the form

$$k(x,u) \;=\; \frac{1}{(2\pi)^{\frac{d+1}{4}}|\Sigma|^{\frac{1}{4}}}\;\text{erf}(u^T\tilde{x})\;\exp(\frac{-u^T\Sigma^{-1}u}{4}), \tag{9}$$

where d is the dimensionality of the input data (2 for x,y). 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 ($Y_i(s)$). Equations 10 and 11 show the mathematical formulation of the process convolution approach.

$$Y_i(s) \;=\; U_i(s) \;+\; W_i(s) \tag{10}$$

$$U_i(s) \;=\; \int_s k_i(s,\lambda) \;\star\; X(\lambda)\; d\lambda \tag{11}$$

Fusion GP regression takes into account points 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 [24]. Boyle et al. apply this technique for stationary squared exponential kernel. The cross covariance function is given by Equation 12 and the resulting auto covariance function is specified in Equation 13.

$$K_{ij}^U(x,x') \;=\; K_f(i,j)\,(2\pi)^{\frac{d}{2}}\,|\Sigma_i \;+\; \Sigma_j|^{-\frac{1}{2}}\;\exp\left(-\frac{1}{2}(x-x')^T\Sigma_{ij}(x-x')\right) \tag{12}$$

where

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

$$K_{ii}^U(x,x') \;=\; K_f(i,i)\,(\pi)^{\frac{d}{2}}\,|\Sigma_i|^{-\frac{1}{2}}\;\exp\left(-\frac{1}{4}(x-x')^T\Sigma_i(x-x')\right) \tag{13}$$

The K_f terms in Equations 12 and 13 are inspired by [23]. 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.

This work inspired from [25] and [24] 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 cross covariance is specified by Equation 14.

$$K_{ij}^U(x,x') \;=\; K_f(i,j)\,2^{\frac{d+1}{2}}\,|\Sigma_i|^{\frac{1}{4}}\,|\Sigma_i \;+\; \Sigma_j|^{-\frac{1}{2}}\,|\Sigma_j|^{\frac{1}{4}}\,k_{NN}(\mathbf{x},\mathbf{x}',\Sigma_{ij}) \tag{14}$$

where

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

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 2. The resulting auto-covariance function for the NN kernel is given by Equation 15.

$$K_{ii}^{U}(x, x') \;=\; K_f(i, i)\, k_{NN}(\mathbf{x}, \mathbf{x}', \Sigma_i) \tag{15}$$

As before, the K_f terms in Equations 14 and 15 models the task similarity between individual data sets. It is a symmetric matrix of size nt x nt and is learnt along with the other GP hyperparameters. Thus, the hyperparameters of the system that need to be learnt include $(nt.(nt+1))/2$ task similarity values, $nt\,.\,2$ or $nt\,.\,3$ length scale values respectively for the individual SQEXP or NN kernels and nt noise values corresponding to the noise in the observed data sets.

3.5 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 [3], 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.

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

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

3.6 Subjective comparison of the data fusion approaches

1. The GP approach to data fusion simply adds new data to an existing GP model. The HGP approach assumes that the same GP hyperparameters can be used for all data sets and the differences between them can be sufficiently captured by the noise parameter that is learnt for individual data sets. This assumption can be problematic when data from heterogeneous sensors are fused. The DGP approach learns separate GP hyperparameters as well as noise parameters for each data set and then attempts to model correlations between the individual GP's. Thus, the DGP model provides for maximum model flexibility (and is consequently the most complex), followed by the HGP and lastly, the GP.

2. The DGP approach to data fusion requires the derivation of closed form expressions for the auto and cross covariance functions between the covariance functions of the respective GP's. This is not trivial. Traditionally, DGP's have used the stationary squared exponential kernel which provides a convenient solution. The approach presented in this report derives and uses the required expressions corresponding to the non-stationary neural network kernel.

3. The table below shows the number of hyperparameters that need to be optimized for each approach. Based on this, it would be expected that the susceptibility to optimization issues (poor local optima) and the time required for learning/convergence would increase from the GP (min) to the HGP and the DGP (max) approach.

Model	Task similarity / signal variance	Covariance function NN (SQEXP)	Noise
GP	1	3 (2)	1
HGP	1	3 (2)	nt
DGP	$(nt\,.\,(nt+1))\,/\,2$	$3\,.\,nt$ $(2\,.\,nt)$	nt

4. The HGP (and GP) approach provides for a batch and a recursive estimator [2]. The DGP approach is natively a batch model. The HGP approach can be readily parallelized for several data sets.

5. The DGP approach may more easily cope with registration errors between data sets. This is due to modeling of the data sets using separate GP's.

6. The DGP approach can be used to predict multiple properties of the entity being modeled. An example of this has been demonstrated in [3] where the DGP/MOGP approach is used for simultaneous elevation-color modeling of the terrain. This is a demonstration of data fusion across heterogeneous information *types* (not just heterogeneous sources of the same kind of information).

7. This section described fundamental approaches to data fusion using Gaussian processes. In practice, these basic approaches can be adapted to different situations yielding multiple variants. These are described in Section 4.2 and compared in the experiments presented thereafter.

4 Experiments

Experiments were conducted on multiple and/or multi-sensor data sets taken from mining scenarios. Section 4.1 describes the data sets used. The methods that have been tested, including previously unreported methods, are discussed in Section 4.2. The methodology of testing is then described in Section 4.3. Several metrics have been used to evaluate the methods, these are described in Section 4.4. Results obtained are then presented and discussed in Section 4.5. Outputs of the data fusion process provided by the best performing model (suggested by the evaluation) and the most flexible model are also presented.

4.1 Data sets

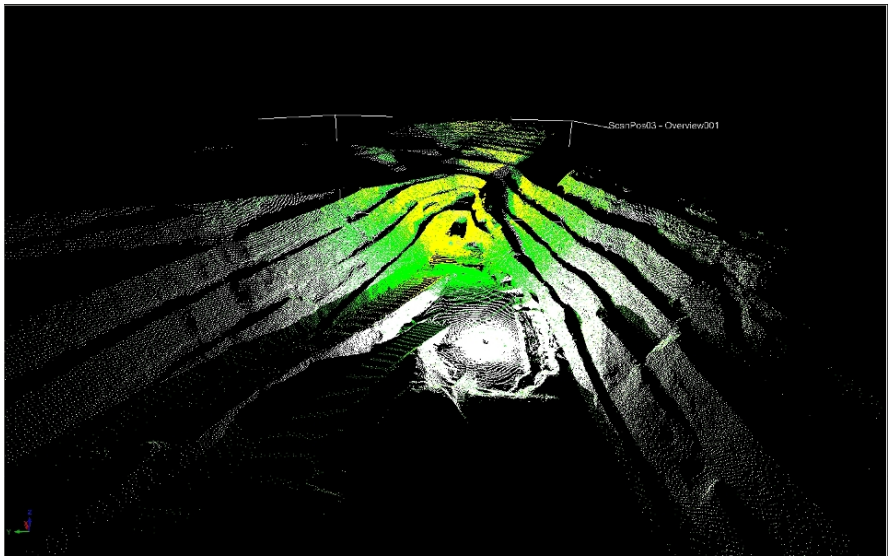

Figure 2: West Angelas data set. Three overlapping RIEGL laser scans. Each scan had on an average about 500,000 points spread over about 1.8 x 0.5 sq km.

Two real sensor data sets taken from mine sites have been used for evaluation of the data fusion approaches. These data sets were collected by professional surveyors in mine sites as part of their routine operations. Data collected by surveyors at mine sites are typically done using high end RTK-GPS systems with very precisely located base stations. This results in positioning accuracies of $\leq$ 10mm in the horizontal direction and $\leq$ 20mm in the vertical direction, during surveying operations. All data are represented with respect to a mine specific coordinate system which serves as the reference for the life of the mine site and all of its operations. Further, data collected during a survey are subject to a registration process, based on a method similar to the Iterative Closest Point algorithm [32], before any further processing is done. Thus, given the positioning accuracies, the span of the data (large sections of mines) and the preliminary registration process, the data registration aspect between scans within a data set is not considered for the purpose of the thesis of this report and its evaluation. Extending the presented approaches to handle data sets with registration errors is discussed in Section 4.5.

Figure 2 depicts a three scan data set from the West Angelas mine in Western Australia. The data set comprises of 3 scans taken using a RIEGL LMSZ420 laser scanner. 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 with

the first two scans being significantly overlapped and the last scan being significantly displaced from the other two. All three scans are densely populated near the point of data collection and become very sparse as one moves away from this point. The outer sections of the scans, in particular, are characterized by large gaps and sparse data. This data set is hereafter referred to as WAMS.

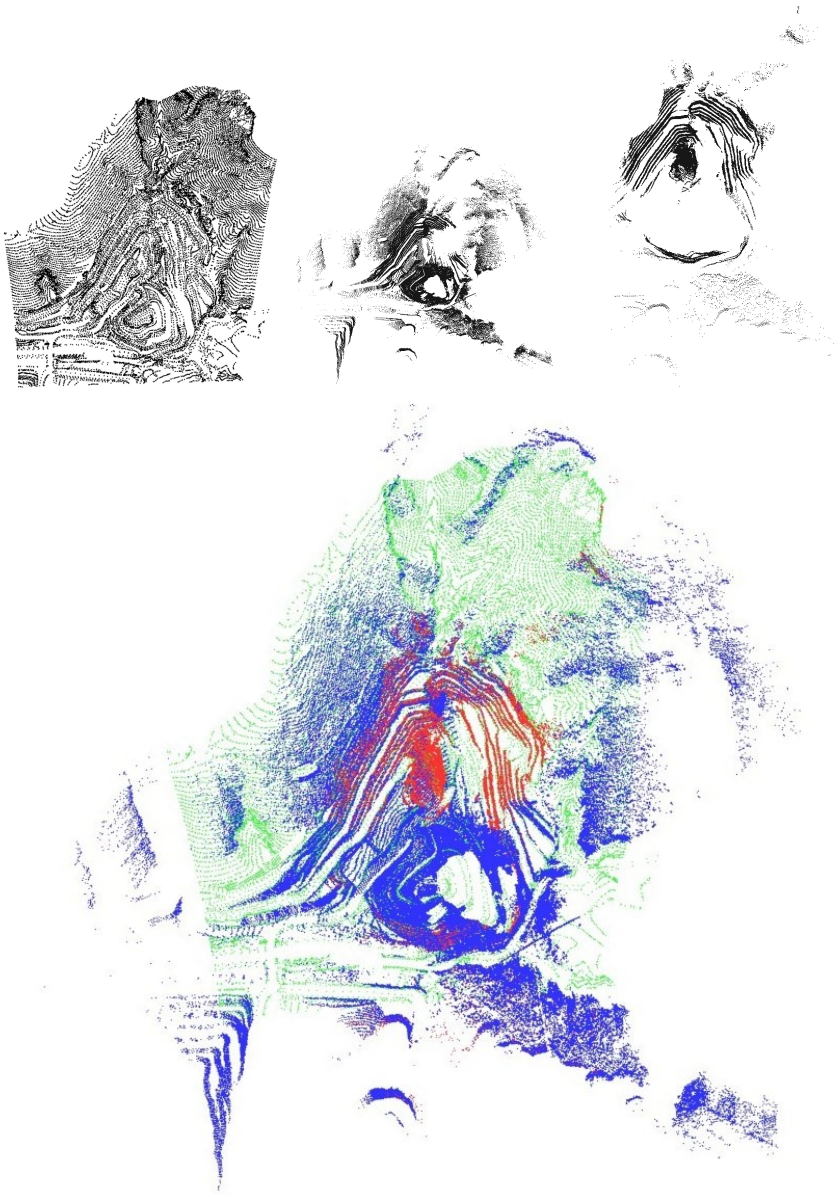

Figure 3: The Mt. Tom Price mine data set. The top row shows the GPS data, wide area laser scan 1 and limited area laser scan 2, in that order, which are respectively shown in green, blue and red in the figure below. The image below is an overlay of the data sets to show the context.

A second data set was used to test GP data fusion approaches using multiple multi-sensor data (RIEGL laser scanner and GPS survey) acquired from a large mine pit. Three surveys of the same region, of differing characteristics, were acquired from Mt. Tom Price mine in Western Australia. The first was a sparse GPS Survey having only about 34,530 points spread over 1437.2 m x 1879.5 m x 380.5 m. The second 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 third data set was a dense (about 400,000 points) RIEGL laser scan spread over a 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. This is a challenging data set. The GPS data and the laser scans have very different characteristics in terms of spread/resolution and accuracy. The two scans themselves are very different in that one spans a very limited area relative to the other and both scans are very sparsely and intermittently populated in the outer sections. This data set is hereafter referred to as TPMM.

4.2 Methods

The three basic approaches to data fusion namely GP, HGP and DGP need to be tested. These basic approaches are compared against their variants and a benchmark naive estimator. The following paragraphs describe the methods and variants tested.

The GP approach uses only one set of hyperparameters and as a result uses only one signal variance hyperparameter for each data set. As an intermediate step in complexity between the GP and DGP approaches, the GP data fusion approach can also be combined with the K_f task similarity parameters that are used in the DGP approach. This approach, denoted hereafter as GP-Kf will also be tested. Likewise, an intermediate step in complexity between the HGP and DGP approaches is the HGP-Kf variant of the basic HGP approach. In the following paragraphs, two separate variants of the DGP approach are discussed.

In each of the methods of data fusion discussed in this report, the vector of observations $\mathbf{z}$ is given by

$$\mathbf{z} = [\ \mathbf{z}_1\ ,\ \mathbf{z}_2\ ,\ \mathbf{z}_3\ ,\ \ldots\ ,\ \mathbf{z}_{nt}\ ]'$$

where $\mathbf{z_i}$ are the elevation data obtained from individual data sets. As mentioned in Section 3.1, the mean function is assumed zero by appropriately shifting/scaling the elevation data. In both simple GP and HGP data fusion, there is only one GP that is regressed. Thus, the mean of the set of observations from all data sets combined is used to make the mean of the GP zero. This is given by

$$\mathbf{z} = [\ \mathbf{z}_1 - \mu\ ,\ \mathbf{z}_2 - \mu\ ,\ \mathbf{z}_3 - \mu\ ,\ \ldots\ ,\ \mathbf{z}_{nt} - \mu\ ]'$$

where $\mu = \frac{\Sigma_{i=0}^{nt} \mathbf{z_i}}{\Sigma_{i=0}^{nt} N_i}$ and N_i is the number of data from the i^{th} data set. In the DGP data fusion case, two possibilities exist - (1) treat the set of GP's as a single output DGP system with different GP hyperparameters for different data sets and mean shift all the data sets together (hereafter denoted as DGP-1) and (2) treat the individual GP's independently and mean-shift each data set by its mean (hereafter denoted as DGP-M). Mathematically, the first option uses the exact same expression above and the second option is expressed as follows - for $\mu_i = \frac{\Sigma_{i=0}^{nt} \mathbf{z_i}}{N_i}$,

$$\mathbf{z} = [\ \mathbf{z}_1 - \mu_1\ ,\ \mathbf{z}_2 - \mu_2\ ,\ \mathbf{z}_3 - \mu_3\ ,\ \ldots\ ,\ \mathbf{z}_{nt} - \mu_{nt}\ ]'$$

The justification for the use of the first option (DGP-1) is that the same quantity (ie. one output or task) is being modeled. This option can be used when the data sets are similar (ie. similar span, similar resolution, similar accuracy) and accurately registered. In this case, the mean over all data sets can significantly change (ideally, reduce) the error after data fusion. This option also allows for an exact comparison of the DGP approach with the other two approaches.

However, when the data sets are very different, it may be beneficial to treat them as different GP's and the data fusion objective should be to improve one GP's output given its correlation with other data sets. The improvement may be less pronounced than in the previous case as the other GP's / data sets serve to only "inspire" the GP in consideration. This is because irrespective of how many data sets are used, the mean for a GP to be regressed will remain constant and the change will only be in the "residual", using those from other data sets.

As a benchmark naive estimator, a Gaussian distribution is fit to the local neighborhood data used for the test point regression process. The predicted elevation is the mean of the local elevation values and the predicted variance is the variance of the local elevation values. This method could for example be applied to data fusion in case of grid/elevation maps. Thus, the methods in comparison are DGP-M, DGP-1, HGP-Kf, GP-Kf, HGP, GP and the naive approach.

4.3 Testing procedure

The objective of the experiment was to compare the GP, HGP and DGP approaches to data fusion and their variants using the nonstationary NN and stationary SQEXP kernel and both the WAMS and TPMM data sets. A ten fold cross validation was performed with each data set being tested individually and with each kernel. These are denoted in the tables that follow by NN-WAMS (NN kernel applied to WAMS data), SQEXP-WAMS, NN-TPMM and SQEXP-TPMM.

The benchmarking experiment presented in this report is an *exact* comparison between the aforementioned approaches. To do this,

- The best available DGP-M (most general approach) parameters were found for each kernel applied to each data set. From this, appropriate subsets of the parameters were chosen for the other simpler approaches.

- The approaches were compared on identical test points and identical training/evaluation points selected for each of the test points.
- It is also necessary that the covariance function for the simple GP / HGP approaches *must* be identical to the auto-covariance function of the DGP approach. For this reason, the auto-covariance function (for both kernels) is used as the covariance function for the GP / HGP approaches to data fusion.

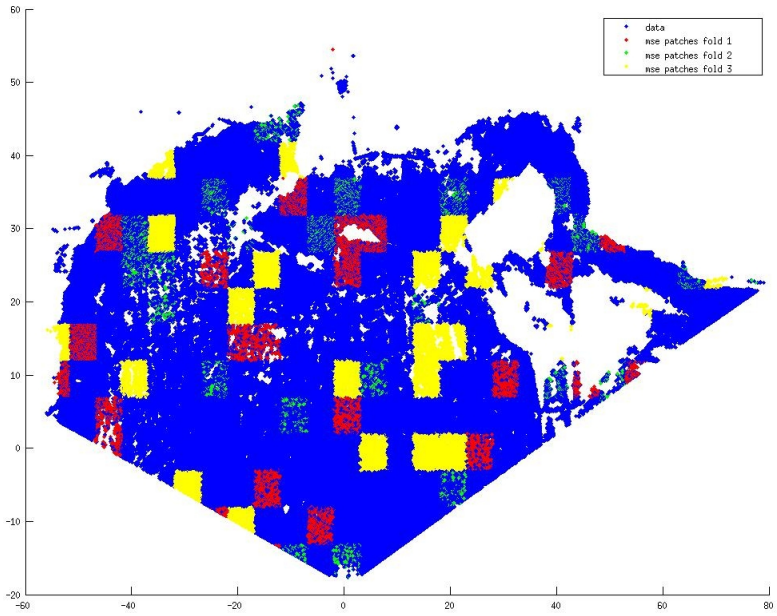

Figure 4: Example of patch sampling of a single 135m x 72m random scan into patches of size 5m. Three of the ten folds are shown, each with a separate color. The experiments in this report use significantly larger data sets and correspondingly larger patches of size 50m. Test points within these patches have "support" data away from them, outside the patches. The sampling method is therefore a stronger test of the robustness of an approach to estimating the elevation at the test point. The estimation errors though, will be higher than that obtained for a uniformly sampled set of points.

For the cross validation, a "patch" sampling technique was used as in [1] (see Figure 4). The idea was that rather than selecting points uniformly, patches of data test the robustness of the approach better as the support points to the query point are situated farther away than in uniform point selection. The data set is gridded into 50 m x 50 m patches. Collections of patches represent individual folds. In each cross validation test, one fold was designated as a test fold and test points selected from it were used exclusively for testing only. All other folds together constituted the evaluation data, a small subset of which were labeled as the training data. Note that this technique of testing will naturally lead to larger errors. From each test fold, 10,000 test points (or the maximum available) were selected as 200 patches of 50 points each. The elevation estimates (and error metrics defined in the following section) with a single data-set alone are computed. Subsequently, the same estimates are computed when fusing this data set with one other data set alone and thereafter with all three data sets combined. The result of a 10 fold cross validation test is a 100,000 point evaluation in tougher test conditions than what would be obtainable with uniform sampling. Various metrics (see Section 4.4) have been used in evaluating the approaches. The criteria for evaluating the occurrence of data fusion was absence of test cases exhibiting an increase in uncertainty. A decreasing trend for other metrics would justify the data fusion as being useful in the context ie. the model/method in question is able to cope with the data to be fused. Note that the order of the fusion of data sets does not change either the method or the results.

4.4 Metrics

Multiple metrics have been used to understand the various methods being tested. They are briefly described below. These are evaluated for each test point in each fold of the cross validation test. The result would then be represented by the mean and standard deviations of all values across all folds (100,000 test cases in 10 folds).

1. *Squared Error (SE)* This represents the squared difference between the predicted elevation and the known elevations for the set of test points. The mean over the set of all test points (Mean Squared Error or MSE) is the most popular metric for the context of this report. Referring Equations 4 and 5, for the i^{th} test point,
$$SE(i) \ = \ (\bar{f}_*(i) - z_i)^2$$

2. *Standardized squared error (SSE)* The SE/MSE is sensitive to the scale of the data and since the cross validation involves random point selections from different parts of the data set, this assumes significance. Inspired by [11] (see page 23), the SSE computes the ratio of the SE and the variance of the local elevation values ie. the variance of the neighborhood evaluation points that are used in the regression for the particular test point. Ideally, the SSE should decrease with each fusion step. Referring Equations 4 and 5, for the i^{th} test point,
$$SSE(i) \ = \ \frac{(\bar{f}_*(i) - z_i)^2}{var(\mathbf{z})}$$
where $\mathbf{z}$ represents the vector of elevation values of the neighborhood points that are used to estimate the elevation at test point i.

3. *Negative log probability / Log loss (NLP)* Inspired by [11] (see page 23), this is a measure of the extent to which the model (including the xGP model, kernel, parameters and evaluation data) explain the current test point. Unlike in [11], the values are not standardized here as the values of this metric from a simple Gaussian are explicitly provided by the naive estimator for comparison. However, just like in [11], the lower the value (typically much smaller than that of the naive method) of this metric, the better the model. For data fusion, the desirable outcome would be a decrease in NLP with each step of the fusion, although it is more important to compare this metric across approaches (lower value implies better data fusion approach) for a given extent of data fusion. For the i^{th} test point,
$$NLP(i) \ = \ \frac{1}{2} log(2\pi\sigma_*^2) + \frac{(\bar{f}_*(i) - z_i)^2}{2\sigma_*(i)^2}$$

4. *Change in predicted variance/uncertainty (DVAR)* This represents the change in predicted variance across each data fusion step. For data fusion to occur, the predicted variance must not increase across fusion steps. For the i^{th} test point,
$$DVAR(i) \ = \ \sigma_*(i)_2^2 - \sigma_*(i)_1^2$$
where the subscript represents the fusion sequence step.

4.5 Results and discussion

Tables 1 and 2 show the cross validation results obtained for the WAMS data set using the NN and SQEXP kernels respectively. The corresponding results for the TPMM data set are shown in Tables 3 and 4. The following points may be noted from the experiment and the results obtained.

1. These results represent an exact comparison of several techniques for one set of local optima parameters. It is quite possible that further optimization may lead to better results for the DGP-M and consequently for all other methods. It is also very plausible that separate optimization of the simpler techniques (eg. HGP-Kf) suggested by the experiments would actually produce better parameters and consequently better error estimates than those reported in these tables, due to a significant reduction in optimization complexity. *The fact that simpler techniques can outperform the most general technique for a parameter set optimized to the latter, proves that there exists at least one set of parameters in the parameter-space of the simpler technique for which it can outperform the most general technique (DGP-M)* for the given data set and metrics. Note that these metrics are evaluated over patches of test data and thus are expected to be higher than if evaluated at individually selected test points.

2. It is clear from each test with either kernel that the most general technique, the DGP-M is certainly not the best performing one for the given data set evaluated using the metrics used. For simpler (same sensor, overlapping) data sets such as the WAMS data the difference between the best performing method and the DGP-M is quite significant whereas in more complex cases, the DGP-M is quite competitive with the best performing method. This suggests that for a data set like WAMS, simpler techniques are probably more effective whereas for more complex data sets, a more complex approach is likely to be justifiable as the method of choice.

3. For both data sets, the GP models based on the NN kernel performed better than the corresponding ones using the SQEXP kernel. Experience has demonstrated a better and faster rate of convergence to a reasonable/good set of parameters with the NN kernel than the SQEXP kernel.

4. In all tests, there were no cases of increase in uncertainty (predicted variance) for the GP based approaches. The naive estimator on the other hand displayed significant increases in this respect.

5. The WAMS data set has three registered scans (denoted here as D1, D2, D3) including two significantly overlapping scans and another scan taken farther away. Consequently, for a test point in D1, points from D2 would have roughly similar local neighborhood resolution (ie. length scales in GP terminology) and "elevation neighborhoods" whereas those from D3 will not. Thus, the naive estimator produces a decrease in most error metrics for the first data fusion step and then produces an increase when the D3 scan is also fused.

6. For the WAMS data set, performance with the NN kernel is significantly better than the SQEXP case. For the NN-WAMS case, on the basis of SE, SSE and NLP metrics, the HGP approach outperforms all other approaches; this, even though the NLP increases from the second to the third step of the fusion sequence. In the case of SQEXP-WAMS, the simpler approaches demonstrate an increase in the SE and a sharp rise in the NLP from the second to the third fusion step; further, the NLP values of the simpler approaches are not competitive when compared with other approaches. In this case, a more complex approach, the HGP-Kf demonstrates a decreasing trend in each of the metrics together with the best NLP values for each step of the fusion process.

7. The TPMM data set comprises of three very different types of data - a uniformly spread out GPS survey (D1), a wide area laser scan (D2) and a limited area scan (D3). Both D2 and D3 are dense in the centre and towards the outer sections become sparse and intermittent (see Figure 3). Closer observation of the test points that showed a large MSE increase revealed that the nearest neighbors from D2 and D3, for test points in the middle and outer sections of D1, are actually spatially far from the test point, have different resolutions as well as have different "elevation neighborhoods". This is in addition to the basic challenges of dealing with differing resolutions and accuracies of the data. Consequently, the naive estimator produces large SE/MSE and DVAR increases in each step of the fusion process.

8. The NN-TPMM results are superior to the SQEXP-TPMM with consistently better results for all metrics. For the NN kernel, The HGP-Kf marginally outperforms the GP-Kf producing the lowest values with a decreasing trend (the HGP produces lower values for the MSE but its SE and NLP both undergo sharp increases between steps 2 and 3). In the case of the SQEXP kernel, the GP-Kf is the clear winner outperforming every other approach. The more generic approaches display error-metric increases whereas the GP-Kf does not, indicating that the hyperparameters for the second and third data sets could be further optimized.

9. On the DVAR metric's behavior: two factors that are critical to model complexity are the numbers and values of the hyperparameters that need to be optimized. Increased values result in lower model complexity as demonstrated in [11] (see chapter 5) as the model is less restrictive. The number of hyperparameters is particularly relevant to model complexity in the context of this report. Increasing this number (as we go from GP to DGP-M) results in increased model complexity and consequently, an increased $|K(X,X)|$ and $\log|K(X,X)|$. In an exact comparison between two models, the predictive variance of a test point (see Equation 5) is differentiated by the second term in the RHS, ie. $K(X_*,X)K(X,X)^{-1}K(X,X_*)$. A higher model complexity will thus result in a lower value of this term resulting in a smaller decrease in uncertainty. Thus, simpler models will tend to demonstrate larger decreases in uncertainty.

10. The various data fusion approaches that have been tested in this report represent increased "degrees of freedom" of the basic Gaussian process model to accommodate for the challenges posed by the data. These additional degrees of modeling freedom may include independent noise and/or kernel hyperparameters as well as the K_f matrix and may include other such parameters.

11. A limitation that was brought out by the experiments is the current local approximation method. For data sets such as the TPMM, in the outer sections, this can lead to selection of nearest neighbors from very different elevation neighborhoods. Preliminary attempts for more intelligent nearest neighbor point selection (using heuristics or entropy based measures) were demonstrated in [2]. An improved local approximation method would provide for significantly improved results - this is the subject of future work.

12. Handling registration errors between data sets using the presented approaches can be done by either using off-the-shelf algorithms such as the Iterative Closest Point (ICP) algorithm [32] or its numerous variants [33] or incorporating and learning the registration information within the data fusion approach. The former approach (ICP) would be used to preprocess the data before performing GP modeling or data-fusion. The latter technique would integrate registration, modeling and data fusion within the same GP framework. It could be performed using any of the approaches presented in this report by appropriately formulating the learning/optimization problem. It is the subject of current research of the author towards extending the results presented in this report.

13. In summary, the experiments have demonstrated that for the given data sets, simpler GP data fusion approaches (with adequate degrees of freedom) could be just as effective or could even outperform more complex/generic approaches, for a range of metrics. The nonstationary NN kernel was found to produce superior results to the stationary SQEXP kernel, further validating conclusions of [1]. The suggested method of choosing a GP data fusion approach would be to first identify the challenges posed by the data sets in question and thereafter incrementally add degrees of freedom to a basic GP data fusion approach, eventually using a more complex DGP based approach when warranted by the complexity of the data. The use of a suitable kernel such as the nonstationary NN kernel, the availability of good quality data and a good local approximation method would significantly aid the modeling and data-fusion processes.

Table 1: Cross validation - NN-WAMS, D1/D2/D3 = laser scan 1/2/3

Method	Fusion sequence	SE (sqm) mean (std)	SSE mean (std)	NLP mean (std)	DVAR (sqm) mean (std)
DGP-M	D1	5.2761 (24.9017)	3.6398 (19.6754)	1.7410 (6.9361)	NA
	D1+D2	2.7244 (15.3729)	1.5751 (10.9674)	0.9009 (5.9367)	-0.1468 (0.2967)
	D1+D2+D3	1.5608 (19.2726)	0.3126 (2.7171)	0.7066 (15.0748)	-0.0879 (0.2337)
DGP-1	D1	5.2761 (24.9017)	3.6398 (19.6754)	1.7410 (6.9361)	NA
	D1+D2	1.4374 (7.4351)	0.0667 (0.2563)	0.5133 (4.7709)	-0.1468 (0.2967)
	D1+D2+D3	0.6135 (16.6125)	0.0291 (0.3955)	0.2901 (14.6629)	-0.0879 (0.2337)
HGP-Kf	D1	5.2761 (24.9017)	3.6398 (19.6754)	1.7410 (6.9361)	NA
	D1+D2	0.9905 (4.6448)	0.0536 (0.2133)	0.4344 (4.8748)	-0.1915 (0.4003)
	D1+D2+D3	0.5352 (9.1979)	0.0271 (0.3042)	0.3698 (9.9323)	-0.0626 (0.1664)
GP-Kf	D1	5.2761 (24.9017)	3.6398 (19.6754)	1.7410 (6.9361)	NA
	D1+D2	0.9754 (4.5788)	0.0550 (0.2359)	0.4584 (5.2016)	-0.1949 (0.4021)
	D1+D2+D3	0.5339 (9.5229)	0.0289 (0.3514)	0.4205 (10.6550)	-0.0619 (0.1658)
HGP	D1	5.2761 (24.9017)	3.6398 (19.6754)	1.7410 (6.9361)	NA
	D1+D2	0.0511 (0.4437)	0.0325 (0.3765)	0.0437 (7.5565)	-0.3352 (0.7046)
	D1+D2+D3	0.0494 (0.4125)	0.0245 (0.4784)	0.2402 (9.0551)	-0.0017 (0.0120)
GP	D1	5.2761 (24.9017)	3.6398 (19.6754)	1.7410 (6.9361)	NA
	D1+D2	0.0544 (0.4940)	0.0372 (0.4464)	0.5151 (12.2465)	-0.3372 (0.7053)
	D1+D2+D3	0.0535 (0.4731)	0.0270 (0.5293)	0.8915 (15.0059)	-0.0016 (0.0119)
NAIVE	D1	17.1206 (55.1722)	6.3816 (27.9138)	4.0266 (14.0972)	NA
	D1+D2	6.2086 (21.1486)	0.3709 (0.5433)	1.7190 (1.1385)	1.8908 (15.6302)
	D1+D2+D3	6.9328 (20.4201)	0.3084 (0.4047)	2.3248 (0.5368)	7.5193 (17.7368)

To further prove the hypothesis of this report, the outcomes of the data fusion approaches for the best performing approach and the most general approach were produced and visually compared. The cross validation experiments suggested that data fusion can be performed for WAMS data set using the HGP approach (using an NN kernel) and the TPMM data set, using the HGP-Kf approach (also using the NN kernel). For the WAMS data set the output from the HGP approach is presented (see Figure 5) along side the DGP-M approach (see Figure 6). For the TPMM data set, the HGP-Kf (see Figure 7) and DGP-M (see Figure 8) approaches are presented. In each case, one million points were evaluated using all three data set within WAMS and TPMM. The outputs show the resulting surface map and uncertainty (standard deviation) of the elevation estimates obtained. The experiments use the NN covariance function and its auto and cross covariance forms. The experiments demonstrate that simpler GP data fusion approaches work just as well as the more complex DGP-M approach.

Table 2: Cross validation - SQEXP-WAMS, D1/D2/D3 = laser scan 1/2/3

Method	Fusion sequence	SE (sqm) mean (std)	SSE mean (std)	NLP mean (std)	DVAR (sqm) mean (std)
DGP-M	D1	24.5784 (111.3587)	13.5961 (136.3495)	1.7609 (6.5622)	NA
	D1+D2	16.5476 (72.8656)	5.3974 (30.4179)	1.6049 (6.2931)	-44.9673 (90.1880)
	D1+D2+D3	13.4320 (129.6939)	1.6304 (9.5943)	2.3779 (7.7619)	-16.5221 (33.3010)
DGP-1	D1	24.5784 (111.3587)	13.5961 (136.3495)	1.7609 (6.5622)	NA
	D1+D2	6.1289 (53.3454)	1.2048 (15.0905)	1.5515 (6.2778)	-44.9673 (90.1880)
	D1+D2+D3	2.8626 (127.2487)	0.1946 (2.9065)	1.3902 (6.3863)	-16.5221 (33.3010)
HGP-Kf	D1	24.5784 (111.3587)	13.5961 (136.3495)	1.7609 (6.5622)	NA
	D1+D2	5.0070 (54.6431)	1.1978 (16.5541)	1.5187 (6.3390)	-45.0408 (90.0246)
	D1+D2+D3	1.3709 (115.0489)	0.0914 (1.7884)	1.1719 (6.3625)	-16.0880 (33.1598)
GP-Kf	D1	24.5784 (111.3587)	13.5961 (136.3495)	1.7609 (6.5622)	NA
	D1+D2	5.6077 (98.4109)	1.2978 (17.4780)	1.5341 (6.4344)	-45.1089 (90.0852)
	D1+D2+D3	2.5851 (390.5106)	0.1253 (4.4598)	1.2696 (6.8808)	-16.0997 (33.1699)
HGP	D1	24.5784 (111.3587)	13.5961 (136.3495)	1.7609 (6.5622)	NA
	D1+D2	0.5695 (44.6857)	0.1360 (2.5198)	1.1828 (11.1766)	-63.8789(125.9453)
	D1+D2+D3	0.8169 (103.6789)	0.1036 (3.8702)	1.8022 (12.9168)	-0.2548 (5.3372)
GP	D1	24.5784 (111.3587)	13.5961 (136.3495)	1.7609 (6.5622)	NA
	D1+D2	1.0900 (94.9296)	0.2239 (5.5119)	1.6678 (14.3466)	-63.9392 (126.0189)
	D1+D2+D3	2.3837 (426.6101)	0.2129 (9.9514)	2.4145 (16.2484)	-0.2584 (5.3698)
NAIVE	D1	17.1206 (55.1722)	6.3816 (27.9138)	4.0266 (14.0972)	NA
	D1+D2	6.2086 (21.1486)	0.3709 (0.5433)	1.7190 (1.1385)	1.8908 (15.6302)
	D1+D2+D3	6.9328 (20.4201)	0.3084 (0.4047)	2.3248 (0.5368)	7.5193 (17.7368)

Table 3: Cross validation - NN-TPMM, D1 = GPS survey, D2/D3 = laser scan 1/2

Method	Fusion sequence	SE (sqm) mean (std)	SSE mean (std)	NLP mean (std)	DVAR (sqm) mean (std)
DGP-M	D1	9.1292 (23.8630)	0.2658 (1.4518)	2.4729 (1.1246)	NA
	D1+D2	8.4148 (22.4871)	0.2365 (0.7304)	2.4405 (1.0812)	-0.0770 (0.0746)
	D1+D2+D3	8.2336 (21.9001)	0.2044 (0.6297)	2.4330 (1.0658)	-0.0065 (0.0148)
DGP-1	D1	9.1292 (23.8630)	0.2658 (1.4518)	2.4729 (1.1246)	NA
	D1+D2	8.4097 (22.4688)	0.1547 (0.4207)	2.4403 (1.0810)	-0.0770 (0.0746)
	D1+D2+D3	8.2262 (21.8744)	0.1034 (0.3268)	2.4327 (1.0654)	-0.0065 (0.0148)
HGP-Kf	D1	9.1292 (23.8630)	0.2658 (1.4518)	2.4729 (1.1246)	NA
	D1+D2	8.2607 (22.1759)	0.1522 (0.4161)	2.4333 (1.0744)	-0.1166 (0.1157)
	D1+D2+D3	8.0716 (21.5048)	0.1017 (0.3226)	2.4257 (1.0564)	-0.0058 (0.0141)
GP-Kf	D1	9.1292 (23.8630)	0.2658 (1.4518)	2.4729 (1.1246)	NA
	D1+D2	8.2780 (22.1021)	0.1529 (0.4156)	2.4350 (1.0793)	-0.1289 (0.1230)
	D1+D2+D3	8.0922 (21.4378)	0.1018 (0.3215)	2.4275 (1.0629)	-0.0055 (0.0137)
HGP	D1	9.1292 (23.8630)	0.2658 (1.4518)	2.4729 (1.1246)	NA
	D1+D2	7.5623 (40.5557)	0.1372 (0.5105)	4.7052 (29.1176)	-6.4977 (6.1671)
	D1+D2+D3	8.0768 (63.1431)	0.0917 (0.4702)	11.2188 (88.8729)	-0.3964 (1.8956)
GP	D1	9.1292 (23.8630)	0.2658 (1.4518)	2.4729 (1.1246)	NA
	D1+D2	9.3424 (50.7628)	0.1709 (0.8841)	23.4826 (231.7378)	-6.7084 (6.2689)
	D1+D2+D3	9.7224 (64.0853)	0.1090 (0.6520)	31.1995 (265.6601)	-0.3764 (1.8876)
NAIVE	D1	43.4693 (80.1197)	0.8028 (2.0035)	3.2925 (1.0212)	NA
	D1+D2	85.2614 (211.1752)	0.5565 (0.8507)	3.2743 (0.7543)	38.3761 (201.7045)
	D1+D2+D3	229.4843 (505.2846)	0.5122 (0.6225)	3.7121 (0.7948)	254.9589 (464.0909)

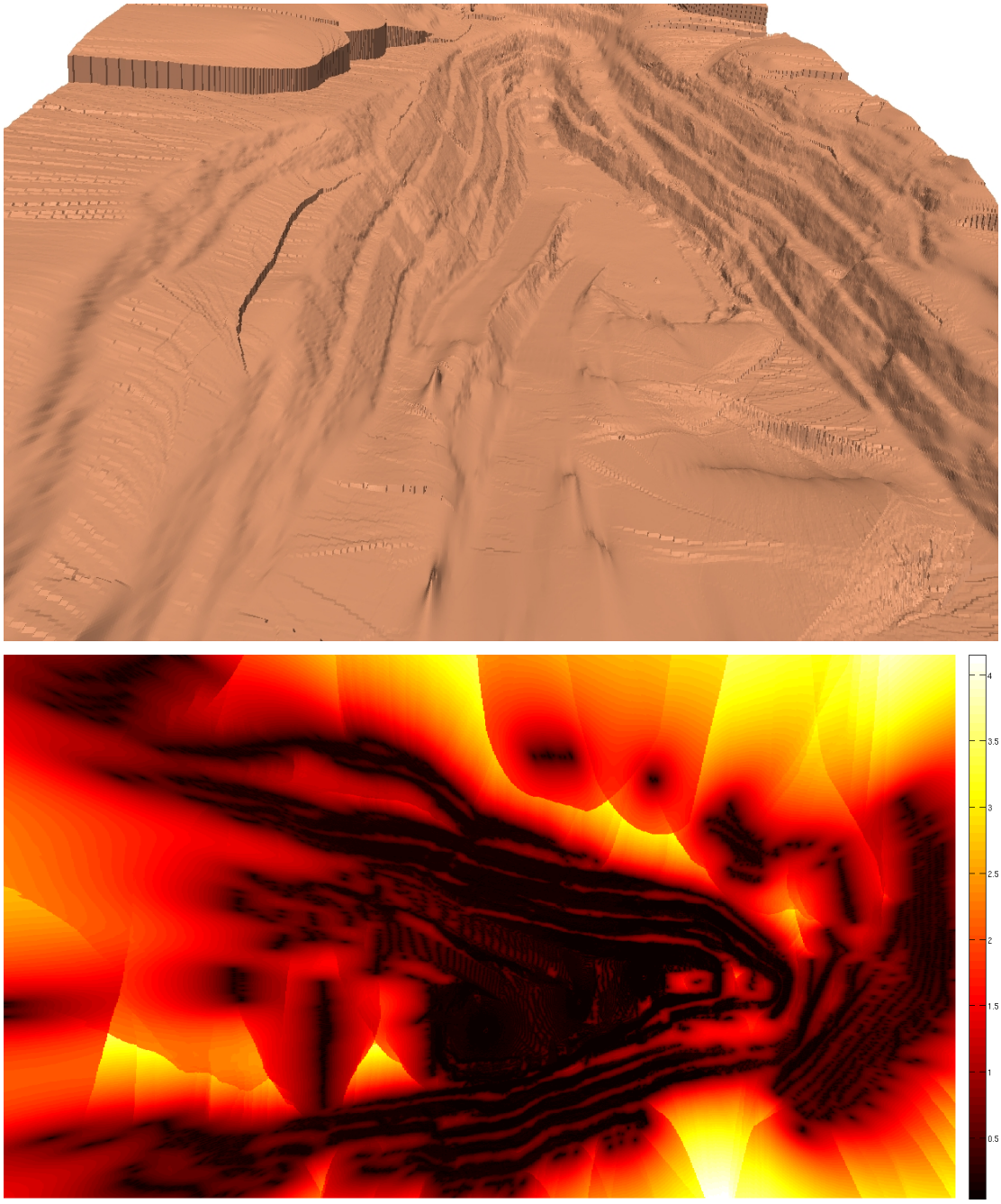

Figure 5: Output of HGP fusion (NN kernel) applied to the West Angelas data sets (three laser scans). The test data comprised of 1 Million points. The figure above shows the surface map produced from the elevation output and the one below represents the uncertainty of the output points of the surface map (standard deviation in m). Note the distinctive step like form on the side walls and the clearly visible roads into the pit.

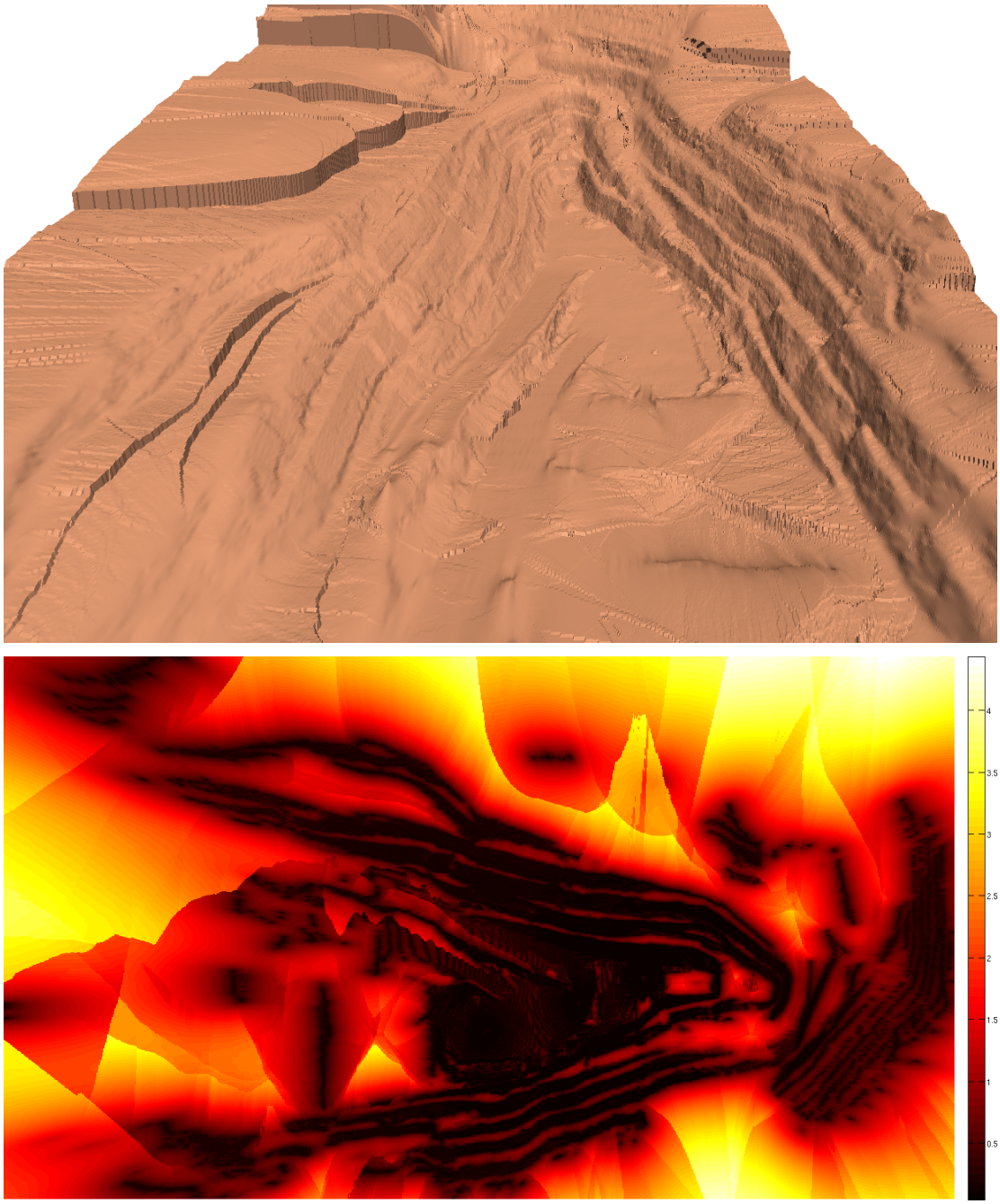

Figure 6: Output of DGP-M fusion (NN kernel) applied to the West Angelas data sets (three laser scans). The test data comprised of 1 Million points. The figure above shows the surface map produced from the elevation output and the one below depicts the uncertainty of the output points of the surface map (standard deviation in m). Note the distinctive step like form on the side walls and the clearly visible roads into the pit.

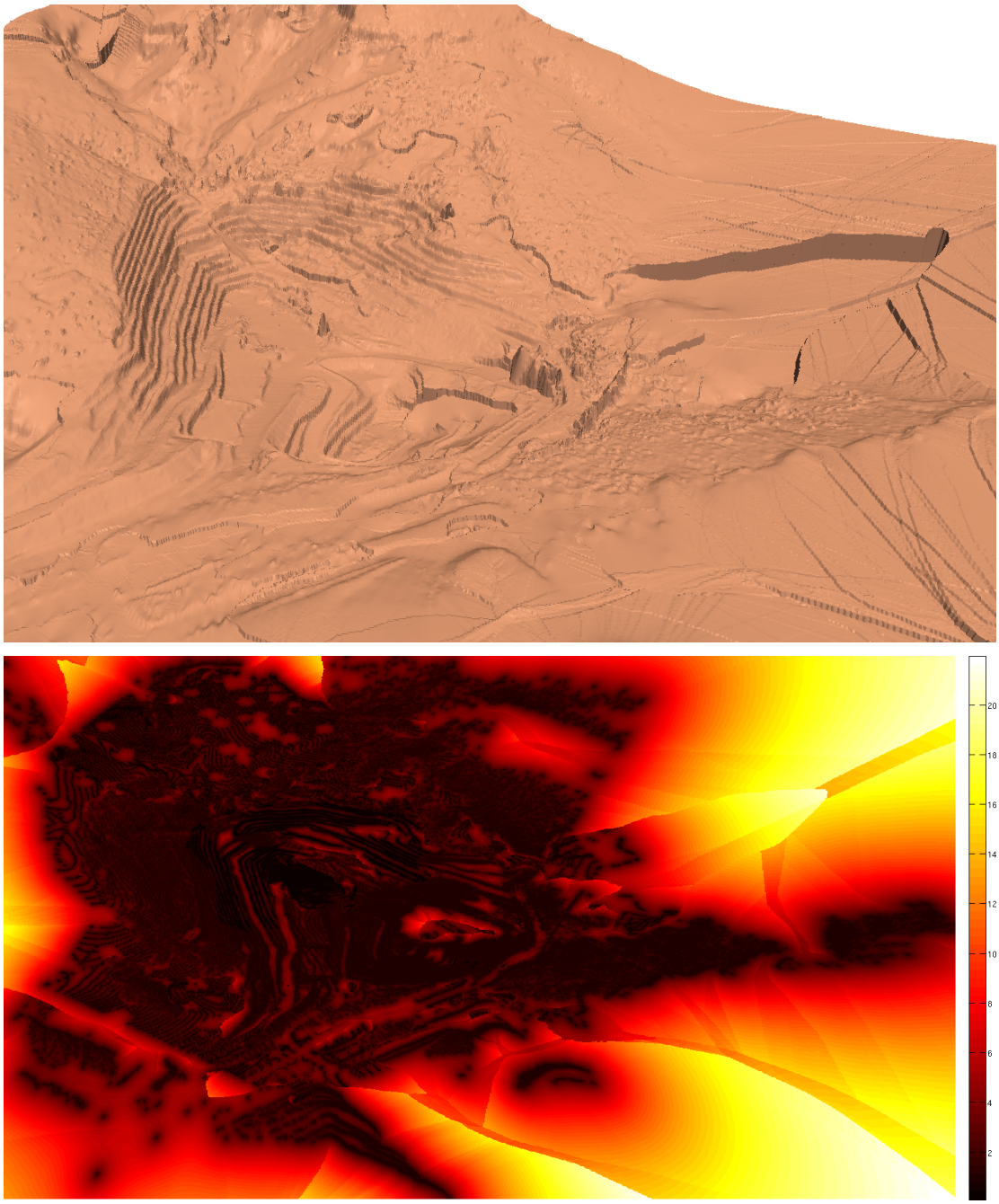

Figure 7: Output of HGP-Kf fusion (NN kernel) applied to the Mt. Tom Price data sets (GPS data + two very different laser scans). The test data comprised of 1 Million points. The figure above shows the surface map produced from the elevation output and the one below depicts the uncertainty of the output points of the surface map (standard deviation in m).

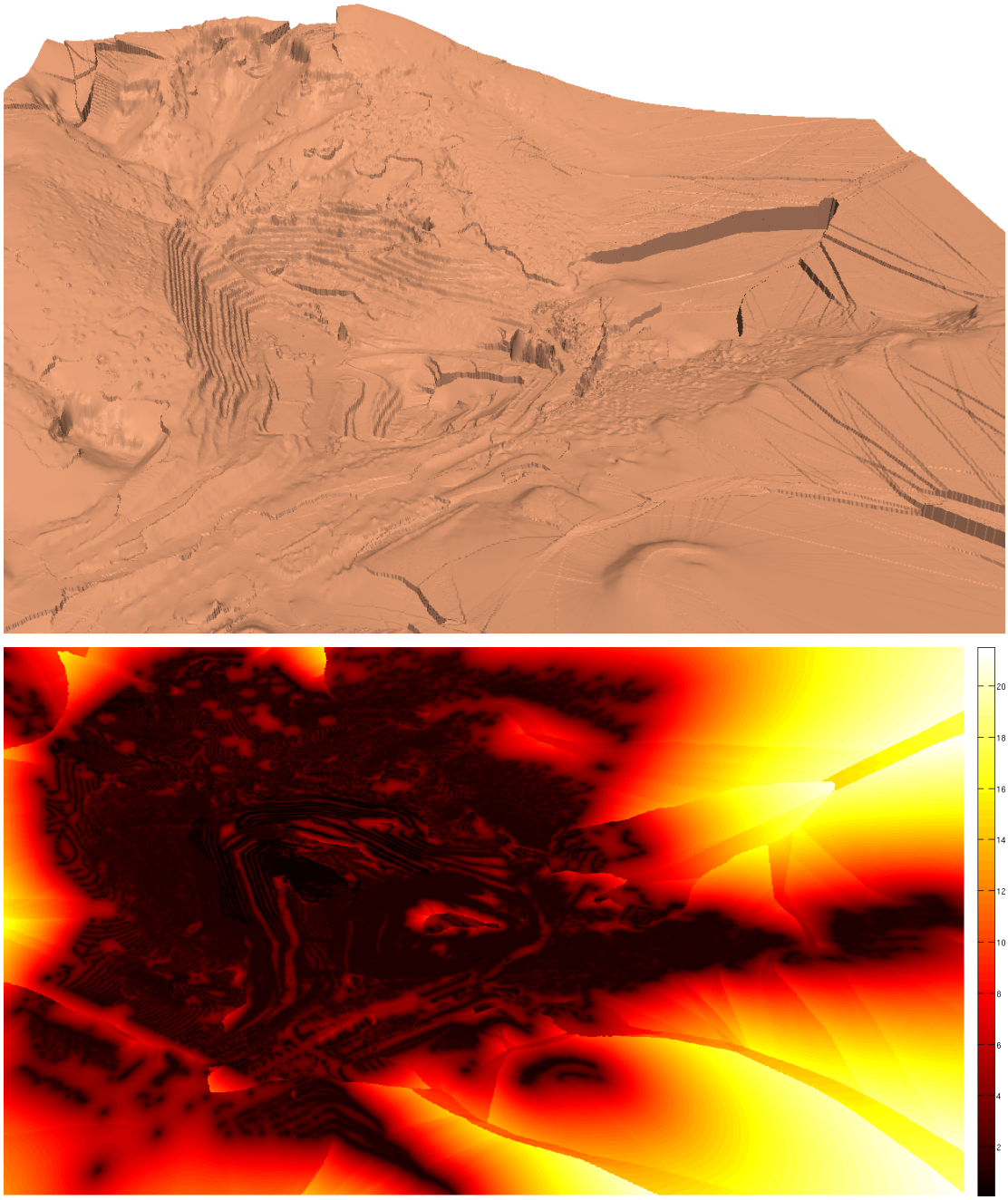

Figure 8: Output of DGP-M fusion (NN kernel) applied to the Mt. Tom Price data sets (GPS data + two very different laser scans). The test data comprised of 1 Million points. The figure above shows the surface map produced from the elevation output and the one below depicts the uncertainty of the output points of the surface map (standard deviation in m).

Table 4: Cross validation - SQEXP-TPMM, D1 = GPS survey, D2/D3 = laser scan 1/2

Method	Fusion sequence	SE (sqm) mean (std)	SSE mean (std)	NLP mean (std)	DVAR (sqm) mean (std)
DGP-M	D1	12.5181 (40.4022)	0.5607 (5.5308)	2.6251 (2.0737)	NA
	D1+D2	11.5673 (36.9398)	0.3272 (1.1606)	2.6041 (2.1341)	-1.7902 (6.4846)
	D1+D2+D3	11.6920 (37.1531)	0.2858 (0.8797)	2.6103 (2.1442)	-0.0715 (0.5226)
DGP-1	D1	12.5181 (40.4022)	0.5607 (5.5308)	2.6251 (2.0737)	NA
	D1+D2	11.5602 (37.1427)	0.2264 (0.7441)	2.6033 (2.1402)	-1.7902 (6.4846)
	D1+D2+D3	11.6432 (36.8965)	0.1457 (0.4851)	2.6073 (2.1408)	-0.0715 (0.5226)
HGP-Kf	D1	12.5181 (40.4022)	0.5607 (5.5308)	2.6251 (2.0737)	NA
	D1+D2	11.5435 (36.8259)	0.2286 (0.8325)	2.6040 (2.1514)	-2.0069 (7.4420)
	D1+D2+D3	11.6586 (36.7451)	0.1483 (0.5169)	2.6144 (2.1867)	-0.1019 (0.6582)
GP-Kf	D1	12.5181 (40.4022)	0.5607 (5.5308)	2.6251 (2.0737)	NA
	D1+D2	11.2344 (35.2670)	0.2214 (0.7294)	2.5918 (2.1181)	-1.8684 (7.1915)
	D1+D2+D3	11.1484 (34.5814)	0.1414 (0.4672)	2.5890 (2.1071)	-0.0871 (0.6192)
HGP	D1	12.5181 (40.4022)	0.5607 (5.5308)	2.6251 (2.0737)	NA
	D1+D2	11.8757 (59.9382)	0.2216 (1.4489)	3.0509 (6.9800)	-7.0732 (17.4067)
	D1+D2+D3	21.8567 (261.8969)	0.1681 (1.2207)	4.7303 (36.0616)	-0.2454 (1.1882)
GP	D1	12.5181 (40.4022)	0.5607 (5.5308)	2.6251 (2.0737)	NA
	D1+D2	10.1228 (47.0318)	0.1856 (0.9906)	2.7857 (5.0904)	-6.7990 (17.0404)
	D1+D2+D3	10.7346 (51.3814)	0.1147 (0.6815)	2.9699 (6.3622)	-0.2121 (1.0664)
NAIVE	D1	43.4693 (80.1197)	0.8028 (2.0035)	3.2925 (1.0212)	NA
	D1+D2	85.2614 (211.1752)	0.5565 (0.8507)	3.2743 (0.7543)	38.3761 (201.7045)
	D1+D2+D3	229.4843 (505.2846)	0.5122 (0.6225)	3.7121 (0.7948)	254.9589 (464.0909)

5 Conclusion

This report presented the state-of-the-art in data fusion using Gaussian processes (GP's). The key contributions of this report are insights on how the various approaches to GP data fusion relate to each other, the effect of model complexity on data fusion as well as how to choose a GP data fusion approach for a particular context. Towards this, three basic data fusion approaches (GP's, heteroscedastic GP's and Dependent GP's) were expressed in terms of the basic Gaussian process model and its regression equations. Multiple previously unreported variants of these approaches were also discussed. The report thus demonstrated multiple generic methods of performing GP data fusion which could be used in any context or application domain. These approaches were subject to a statistically representative cross validation analysis that performed an exact comparison of these models. These experiments also compared the performance of the stationary squared exponential kernel (SQEXP) with the nonstationary neural network kernel (NN) for the data fusion task. Real sensor data (GPS surveys and laser scans) taken from multiple mining scenarios were used in the experiments. The scale of the experiments represents a distinctive feature of this work. The experiments demonstrated that depending on the complexity of the data set in question, simpler GP data fusion approaches could be just as effective or even outperform more complex/generic approaches, for a range of metrics. The NN kernel was found to produce superior results to the SQEXP kernel. The suggested method of choosing a GP data fusion approach would be to first identify the challenges posed by the data and then incrementally add degrees of freedom to the basic GP data fusion approach, eventually using a more complex Dependent GP based approach when warranted by the complexity of the data. The use of a suitable kernel such as the NN kernel and the availability of good quality data and a good local approximation method would significantly aid the process.

A Derivation of the cross-covariance function for the neural network kernel

The process convolution approach [25] 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 [24] 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 [25] and [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 $\int \mid k_i(x_a, \alpha) \mid^2 d\alpha \ < \infty$ 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

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

This can be evaluated analytically (see appendix of [28]) 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

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

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

where

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

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

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

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

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_{ij}^U are given by Equation 26. Consider the expression

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

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

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

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,

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

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, H. Durrant-Whyte, Gaussian Process Modeling of Large Scale Terrain, Journal of Field Robotics 26(10) (2009) 812–840.

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

[3] S. Vasudevan, F. Ramos, E. Nettleton, H. Durrant-Whyte, 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, 2010.

[4] S. Vasudevan, F. Ramos, E. Nettleton, H. Durrant-Whyte, Non-stationary dependent Gaussian processes for data fusion in large scale terrain modeling, in: in the proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 2011.

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

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

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

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

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

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

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

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

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

[14] G. Matheron, Principles of Geostatistics, Economic Geology 58 (1963) 1246–1266.

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

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

[17] M. Girolami, Bayesian data fusion with gaussian process priors: An application to protein fold recognition, in: Workshop on Probabilistic Modeling and Machine Learning in Structural and Systems Biology (PMSB), 2006.

[18] S. Reece, S. Roberts, D. Nicholson, C. Lloyd, Determining intent using hard/soft data and gaussian process classifiers, in: Proceedings of the 14th International Conference on Information Fusion (FUSION), 2011.

[19] P. W. Goldberg, C. K. I. Williams, C. M. Bishop, Regression with Input-dependent Noise: A Gaussian Process Treatment, in: M. I. Jordan, M. J. Kearns, S. A. Solla, L. Erlbaum (Eds.), Advances in Neural Information Processing Systems (NIPS) 10, MIT Press, Cambridge, MA, 1998.

[20] M. Yuan, G. Wabha, Doubly Penalized Likelihood Estimator in Heteroscedastic Regression, Tech. rep., Department of Statistics, University of Wisconsin, Madison, WI, USA (2004).

[21] Q. Le, A. Smola, S. Canu, Heteroscedastic Gaussian Process Regression, in: International Conference on Machine Learning (ICML), 2005.

[22] K. Kersting, C. Plagemann, P. Pfaff, W. Burgard, Most Likely Hetereoscedastic Gaussian Process Regression, in: International Conference on Machine Learning (ICML), 2007.

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

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

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

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

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

[28] C. K. I. Williams, Computation with infinite neural networks, Neural Computation 10(5) (1998) 1203–1216.

[29] C. K. I. Williams, Prediction with Gaussian processes: From linear regression to linear prediction and beyond, in: M. I. Jordan (Ed.), Learning in Graphical Models, Springer, 1998, pp. 599–622.

[30] K. Hornik, Some new results on neural network approximation, Neural Networks 6(8) (1993) 1069–1072.

[31] A. Melkumyan, F. Ramos, Multi-Kernel Gaussian Processes, in: in the proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 2011.

[32] P. Besl, N. McKay, A method for registration of 3-d shapes, IEEE Transactions on pattern analysis and machine intelligence 14 (2) (1992) 239–256.

[33] S. Rusinkiewicz, M. Levoy, Efficient variants of the icp algorithm, in: 3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on, IEEE, 2001, pp. 145–152.