Robot Learning

A Mine on Its Own

Fully Autonomous, Remotely Operated Mine

© STOCKBYTE, EYEWIRE, DIGITAL VISION & BRAND X PICTURES

This study is part of a larger program that is developing and implementing the vision of a fully autonomous, remotely operated mine. A key element of this vision is a robust, probabilistic method of modeling complex terrain on a large scale, typically with dimensions measuring many kilometers. This article presents a study of Gaussian process (GP) models applied to the problems of modeling and data fusion in the context of large-scale terrain modeling. The proposed model naturally provides a multiresolution representation of space, incorporates and handles uncertainties aptly, and copes with incompleteness of sensory information. These attributes are considered essential to support most field robotics applications, including autonomous mining. GP regression techniques are applied to estimate and interpolate (to fill gaps in occluded areas) elevation information across the field. GP approximation methods are introduced to enable the application of the proposed techniques to large data sets. To obtain a comprehensive model of complex terrain, typically, multiple sensory modalities and multiple data sets are required. The GP modeling approach is consequently extended to fuse multiple, multimodal data sets to obtain a best estimate of the elevation given the individual data sets. Two different GP-based concepts are applied to perform data fusion—heteroscedastic GPs and dependent GPs (DGPs). Thus, this article presents a report on an ongoing study of the use of GPs and several GP-based concepts to the problem of large-scale terrain modeling in the context of mining automation.

Contemporary surface mapping methods employ representations based on tessellations. However, this process does not have a statistically sound way of incorporating and managing uncertainty. The assumption of statistically independent data is a further limitation of many works that have used these approaches. Although there are several interpolation techniques known, the independence assumption can lead to simplistic (like simple averaging) techniques that result in inaccurate modeling of the terrain. Thus, an appropriate method for incorporation and management of uncertainty and the ability of handling spatially correlated data are two key challenges on the modeling front.

Typically, sensory data are incomplete because of the presence of entities that occlude the sensor view. This is compounded by the fact that every sensor has a limited perceptual capability, i.e., a limited range and limited 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 (accuracies) and yield data with different levels of sparseness. The problem is in fusing these multiple and multimodal sensory data sets. Terrain data can be obtained using numerous sensors, including three-dimensional (3-D) laser scanners and global positioning system (GPS). The 3-D laser scanners provide dense and accurate data, whereas a GPS-based survey typically comprises a relatively sparse set of well-chosen points of interest. The experiments reported here use data sets obtained from both of these sensors to develop an integrated picture of the terrain.

Digital Object Identifier 10.1109/MRA.2010.936960

BY SHRIHARI VASUDEVAN, FABIO RAMOS, ERIC NETTLETON, AND HUGH DURRANT-WHYTE

1070-9932/10/$26.00©2010 IEEE

GPs provide a powerful, nonparametric, and continuous model for incomplete, uncertain, and correlated data.

In this article, a novel approach to terrain modeling and terrain data fusion using state-of-the-art machine learning methods is presented. A single nonstationary-kernel [neural network (NN)] GP is successfully able to model large-scale terrain data, taking into account the local smoothness while preserving spatial features in the terrain (see the "Gaussian Process Terrain Modeling" section). Particular emphasis is placed on the scalability of the algorithm to large-scale terrain. Toward this, a local approximation method based on a moving-window or a nearest-neighbor methodology, implemented using efficient hierarchical data structures, is proposed (see the "GP Approximations" section). The problem of fusing multiple, multimodal data sets to obtain a comprehensive model of the terrain is then addressed using two different approaches—heteroscedastic GPs (see the "Heteroscedastic GPs for Data Fusion" section) and DGPs (see the "DGPs for Data Fusion" section). Sample outcomes are shown in the respective sections with links being provided to detailed reports on each method introduced in this article. A discussion of current findings and ongoing work concludes this report.

Related Work

The 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 [1], [2], triangulated irregular networks (TINs) [3], [4], contour models, and their variants or combinations [5], [6]. 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 correct way of incorporating and managing uncertainty.

GPs [7] are powerful nonparametric learning techniques that can handle these issues and produce a scalable multiresolution 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. GPs incorporate and handle uncertainty in a statistically sound way and represent spatially correlated data in an appropriate manner. They model and use the spatial correlation of the given data points to estimate the elevation values for other unknown points of interest. In an estimation sense, GPs provide the best linear unbiased estimate [8] based on the underlying stochastic model of the spatial correlation between the data points. They basically perform an interpolation methodology called *Kriging* [9], which is a standard interpolation technique used in the mining industry. Thus, GPs handle both uncertainty and incompleteness effectively.

Recently, GPs have been applied in the context of terrain modeling [10], [11]. Plagemann et al.'s study [10] is based on using a nonstationary equivalent of a stationary-squared exponential (SQEXP) covariance function [12] and incorporating kernel adaptation techniques (also used by [13] in a similar context) to adequately handle both smooth surfaces and inherent (and characteristic) surface discontinuities. It introduces the idea of a hyper-GP using a stationary kernel to predict the most probable length-scale parameters to suit the local structure. It also proposes to model space as an ensemble of GPs to reduce computational complexity. Vasudevan et al. [11] proposes the use of nonstationary NN kernels to model large-scale discontinuous spatial data. It shows that using a suitable nonstationary kernel can directly result in modeling local structure and smoothness. It also proposes a local approximation methodology to address scalability issues relating to the application of this approach to large-scale data sets. This approximation technique is based on an efficient hierarchical representation (K-dimensional tree (KD-tree) [28]) of the data. It also compares performances of GPs based on stationary (SQEXP) and nonstationary (NN) kernels, as well as several other standard interpolation methods applicable to elevation maps and TINs, in the context of large-scale terrain modeling.

This article first presents an overview of [11]. It then extends the GP terrain modeling approach to integrate multiple, multimodal data sets. Two approaches to data fusion are presented: one based on the notion of heteroscedastic GPs and the other based on DGPs. Note that this work develops only the fusion methodology. The registration of individual data sets to a common reference frame is assumed to be given for this work. Two other related articles that attempt the problem of data fusion in the context of GPs include [14] and [15]. Although the former is based on similar assumptions to the fusion approach presented in this study, it bears a hierarchical learning flavor to it in that it essentially demonstrates how a GP can be used to model an expensive process by 1) modeling a GP on an approximate or cheap process and 2) using the many input–output data from the approximate process and the few samples available of the expensive one together to learn a GP for the expensive process. Murray-Smith and Pearlmutter [15] attempts to generalize arbitrary transformations on GP prior through linear transformations. It hints how this framework could be used to introduce heteroscedasticity and how information from different sources could be fused. However, specifics on how the fusion can actually be performed are beyond the scope of this article.

The data fusion approach based on heteroscedastic GPs [16]–[19] treats the problem by combining different noisy data samples of the common entity being modeled. Both [16] and [19] are particularly relevant to this study. They model the noise variance using a separate GP in addition to the GP governing the noise-free output. Goldberg et al. [16] use Markov-chain Monte-Carlo techniques (MCMC) to estimate the posterior noise variance, whereas Kersting et al. [19] propose a maximum-likelihood approach (faster) using an expectation maximization (EM) algorithm-like iterative optimization procedure for computing the noise variances [34]. The fusion approach presented in this article does not use a separate GP to model noise and does not rely on computationally expensive MCMC-based approaches. The approach is tailored toward handling large data sets (approximately 1 million data points per data set) and thus relies heavily on the local approximation methods. It treats individual terrain data sets as homoscedastic in nature, but different data sets considered together form a heteroscedastic system.

The data fusion approach based on DGPs treats the problem as one of 1) modeling each data set using a GP and 2) formulating the data fusion problem as a conditional estimation problem wherein estimation of a GP is improved using information from other GPs—through learning autocovariances and cross-covariances between them. This idea has been inspired by recent machine learning contributions in GP modeling [20], [21], the latter approach being based on [22]. In kriging terminology, this idea is akin to cokriging [23]. This formalism is used to demonstrate multioutput GPs (MOGPs) in the context of simultaneous modeling of both elevation and color of terrain data. It is also used to demonstrate data fusion of multiple, multimodal terrain data sets by casting the problem as a conditional estimation problem, given multiple DGPs.

Experiments have been performed on large-scale multimodal terrain data obtained from real mining scenarios. The scale of the experiments represents a first of its kind in the context of the topic. Toward ensuring the scalability of the approach, approximation methods have been introduced for both the learning and inference stages. This article, thus, summarizes recent studies on using GPs and many of its associated techniques to the problem of large-scale terrain modeling, specifically addressing issues relating to modeling and fusing large-scale terrain data.

GP Terrain Modeling

GPs provide a powerful framework for learning models of spatially correlated and uncertain data. GP regression provides a robust means of estimation and interpolation of elevation information and can handle incomplete sensor data effectively. GPs are nonparametric approaches in that they do not specify an explicit functional model between the input and output. They may be thought of as a Gaussian probability distribution in function space and are characterized by a mean function $m(\mathbf{x})$ and the covariance function $k(\mathbf{x}, \mathbf{x}')$, where

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})], \tag{1}$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))], \tag{2}$$

such that, the GP is written as:

$$f(\mathbf{x}) \sim \mathrm{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \tag{3}$$

The mean and covariance functions together specify a distribution over functions. In the context of terrain modeling, each $\mathbf{x} \equiv (x, y)$ and $f(\mathbf{x}) \equiv z$ of the given data. The covariance function models the relationship between the random variables corresponding to 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. There are numerous covariance functions (kernels) that can be used to model the spatial variation between the data points. The most popular kernel is the SQEXP kernel given as:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \Sigma (\mathbf{x} - \mathbf{x}')\right), \tag{4}$$

where k is the covariance function or kernel, $\Sigma = \left[\begin{smallmatrix} l_x & 0 \\ 0 & l_y \end{smallmatrix}\right]^{-2}$ is the length-scale matrix (a measure of how quickly the modeled function changes in the directions x and y), and 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 NN kernel [24]–[26] is a nonstationary kernel that takes the form:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \arcsin\left(\frac{\beta + 2\mathbf{x}^T \Sigma \mathbf{x}'}{\sqrt{(1 + \beta + 2\mathbf{x}^T \Sigma \mathbf{x})(1 + \beta + 2\mathbf{x}'^T \Sigma \mathbf{x}')}}\right), \tag{5}$$

where β is a bias factor, Σ is the length-scale matrix as described earlier, and $l_x, l_y, \sigma_f, \beta$ constitute the kernel hyperparameters. This kernel represents the covariance function of a NN with a single hidden layer between the input and output, infinitely many hidden nodes and using a sigmoid as the transfer function [25] for the hidden nodes. Hornik [27] showed that such NNs are universal approximators of data, and Neal [24] observed that the functions produced by such a network would tend to a GP.

The main difference between these two kernels is that the SQEXP kernel, being a function of $|\mathbf{x} - \mathbf{x}'|$, is stationary (invariant to translation), whereas the NN function is not so. In practice, the SQEXP function has a smoothing or averaging effect on the data. The NN covariance function proves to be more effective in handling discontinuous (rapidly changing) data than the SQEXP covariance function. This is the main reason why it proves to be effective in modeling complex terrain data. A more detailed analysis of these kernels and their properties is presented in [11].

For dense and large data sets, not all data are used for learning the GP model as such an approach would not scale for reasons of computational complexity. The data are thus sampled (e.g., uniformly) into three disjoint subsets: training, evaluation, and testing. The training data are used to learn the GP model corresponding to the data. The training and evaluation data together are stored in an efficient data structure (a KD-tree [28] is used here) for later use in the inference process. The KD-tree provides for rapid data access in the inference process and is also used in the local approximation methods proposed in the "GP Approximations" section. The testing data are the subset of data over which the GP model is evaluated.

Training the GP for a given data set is equivalent to choosing a kernel function and optimizing the hyperparameters for the chosen kernel. For the SQEXP kernel, this amounts to determining the values $\theta = \{l_x, l_y, \sigma_f, \sigma_n\}$, and for the NN kernel, this amounts to determining the values $\theta = \{l_x, l_y, \sigma_f, \beta, \sigma_n\}$, where σ_n^2 is the noise variance of the data being modeled. This is performed by formulating the

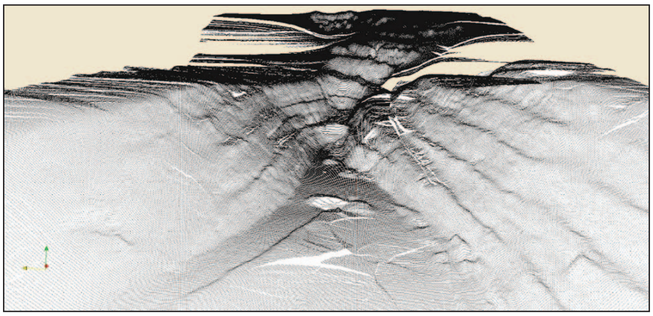

Figure 1. Outcome of applying a single NN-based GP to the West Angelas data set.

problem in a maximum marginal-likelihood estimation framework and subsequently solving a nonconvex optimization problem. This study only considers noise in the observations or output data. Noise in the input data (x, y) may be incorporated as demonstrated in [29].

Applying the GP model amounted to using the learned GP model to estimate the elevation information across a region of interest, characterized by a grid of points at a desired resolution. The 2.5-D elevation map can then be used directly or as a surface map for various applications. This is achieved by performing GP regression at a set of query points, given the training/evaluation data sets and the GP kernel with the learned hyperparameters. GP regression uses the idea that any finite subset of random variables is Gaussian distributed. Thus, any finite set of training or evaluation data and test data are jointly Gaussian distributed. This idea, shown in (6), yields the standard GP regression equations (7) and (8), which represents the posterior/expected-value/mean-value and the variance/uncertainty in the prediction, respectively.

$$\begin{bmatrix} \mathbf{z} \\ f_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right). \tag{6}$$

$$\bar{f}_* = K(X_*,X)[K(X,X) + \sigma_n^2 I]^{-1}\mathbf{z}. \tag{7}$$

$$\begin{aligned}\operatorname{cov}(f_*) = {} & K(X_*,X_*) \\ & - K(X_*,X)[K(X,X) + \sigma_n^2 I]^{-1} K(X,X_*).\end{aligned} \tag{8}$$

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, training outputs $\mathbf{z}$, and the covariance function (kernel), are given by (7) and their uncertainties by (8). The GP estimates obtained are a best linear unbiased estimate for the respective query points. Uncertainty is handled by incorporating the sensor noise model in the training data. The representation produced is multiresolution in that a terrain model can be generated at any desired resolution using the GP regression equations presented earlier. Thus, the GP terrain modeling approach is a probabilistic, multiresolution method that handles spatially correlated information. A detailed report on GP modeling of large-scale terrain data (individual data sets, which may be from any sensor) is presented in [11]. This article includes extensive detail, experiments on multiple, multimodal data sets, as well as comparison and benchmarking experiments. One sample experimental outcome is shown in Figures 1 and 2.

The elevation map in Figure 1 is obtained by predicting the elevation for each point across a test grid of points spanning the data set. The figure clearly demonstrates the ability of the GP to 1) reliably estimate the elevation data in known areas, 2) produce models that take into account the local structure to preserve the characteristics of the terrain being modeled, and 3) be able to model sparse and feature rich data sets.

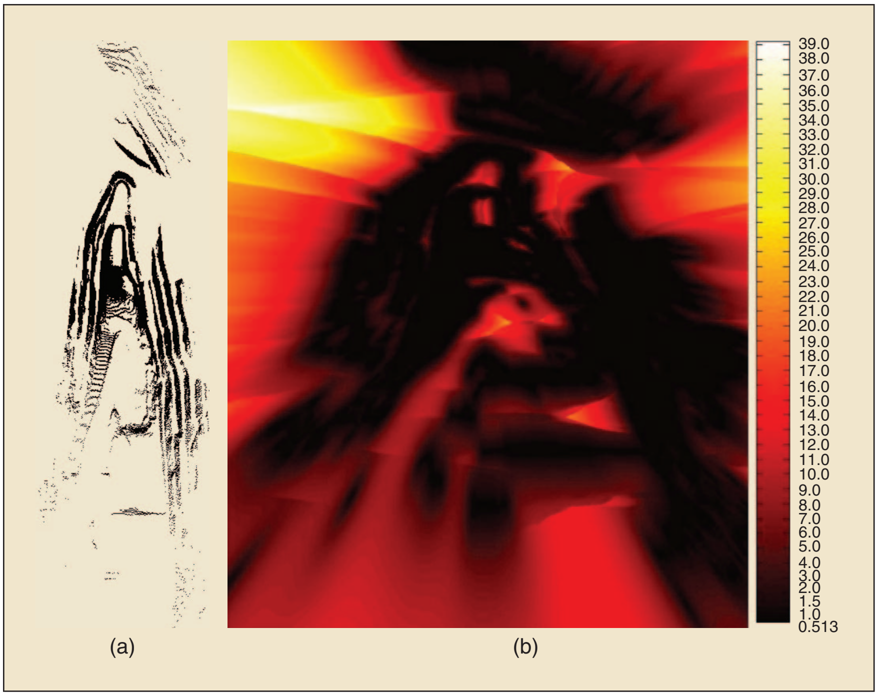

Figure 2. (a) Top–down view of West Angelas mine data set. (b) Uncertainty (standard deviation in meter) of output elevation map (Figure 1).

In Figure 2(b), the final uncertainty estimates of the output elevation map (see Figure 1) of the West Angelas mine data set as obtained using the proposed GP approach are shown. The uncertainty map corresponds to a top–down view of the data set as depicted in Figure 2(a). Each point represents the uncertainty in the estimated elevation of the corresponding point in the elevation map. The uncertainty is clearly higher in those areas where there is no training data—typically in the outer fringe areas and in gaps (minimum standard deviation is 0.01 m, and maximum standard deviation is 39.25 m).

GP Approximations

Both 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. Numerous approximation strategies exist, reviews of which can be found in [7] and [30]. This section introduces a local approximation methodology based on a moving-window or nearest-neighbor methodology that is applied to both GP learning and GP inference.

GP regression techniques are applied to estimate and interpolate elevation information across the field.

Approximation During GP Inference

During the inference process, the KD-tree that initially stores the training and evaluation data is queried to provide a predefined number of spatially closest data, to the point for which the elevation must be estimated. The GP regression process then uses only these training exemplars to estimate the elevation at the point of interest. This approximation method is akin to a moving-window methodology and has multiple benefits that are vital to the scalability of the GP method in a large-scale application. The use of only the spatially closest neighbors for GP regression keeps the computational complexity low and bounded. As described earlier, the SQEXP kernel has a smoothing effect on the data, whereas the NN kernel is more effective in modeling discontinuous terrain data. The number of nearest neighbor exemplars used can be used to control the trade-off between smoothness and feature preservation. Finally, the proposed local approximation method bounds (by selecting a fixed number of data points) the subset of data over which the NN kernel is applied. A more detailed explanation of these aspects is provided in [11]. Vasudevan et al. [31]–[33] deal with the data fusion of multiple large-scale terrain data sets. The proposed approximation method is also used in these articles by extending it to handle multiple data sets for each GP regression performed.

Approximation During GP Learning

Vasudevan et al. [11] demonstrated GP learning for a single large-scale terrain data set. GP learning was based on maximizing the marginal likelihood. This study used uniform sampling to select training points from the data to be modeled as using the several 100,000 data for learning would be computationally infeasible. The same procedure was applicable to the first data fusion work [31] by adopting a constrained optimization approach where GP modeling of only one data set was performed at a time. The recent works on data fusion using DGPs [32], [33] explored the possibilities of jointly learning the GP models of multiple data sets and also learning GP models using only limited computational resources. This was done by applying a nearest-neighbor approximation to GP learning as well.

During learning, a small set of training points is identified through uniform sampling. The KD-tree is also used to select points in each of their neighborhoods as training points. Thus, patches of data are selected for training. Thus, the KD-tree representation of the available data aids in both learning and inference. Once the training data are selected, GP learning proceeds by using the maximum marginal-likelihood framework detailed in [11]. To ensure that GP learning can be effectively performed given limited computational resources, a block-learning procedure was adopted to learn the GP models. Instead of learning with all training points at once, this idea uses blocks of points in a sequential marginal-likelihood computation process within the optimization step. The block size is predefined and depends on the computational resources available. The KD-tree-based block learning guarantees that multiple (in a data fusion context) large data sets can be handled using even limited computing resources. Experiments in [32] demonstrated that for a similar GP learning performance (resulting model error), the KD-tree-based block learning procedure was significantly faster (by 1 h) than a simple uniform sampling approach. Together with the nearest-neighbor/moving-window GP approximation in the inference stage, these methods ensure the scalability of the approach to multiple large data sets.

Heteroscedastic GPs for Data Fusion

This approach to data fusion 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 (see the "GP Terrain Modeling" section) with data having different noise parameters. The formulation is similar to the heteroscedastic GP formulation described in [16] and [19]. A detailed description of the approach is presented in [31], and this section provides an overview of the approach and puts it into perspective with that detailed in the "GP Terrain Modeling" section.

Given multiple data sets (possibly multimodal) of the terrain being modeled, the objective is to estimate the elevation at a point given the various data sets and their respective GPs (kernels and their hyperparameters). This can be specified as

$$\mathbb{E}[\,f_*(\mathbf{X}_*)],\, \mathrm{var}(\,f_*(\mathbf{X}_*))|X_i, \mathbf{z_i}, GP_i, X_*, \tag{9}$$

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).

As in the "GP Terrain Modeling" section, the joint distribution of any finite number of random variables of a GP is Gaussian. Thus, the joint distribution of the training outputs $\mathbf{z}$ and test outputs f_*, can be specified by:

$$\begin{bmatrix} \mathbf{z} \\ f_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K(X,X) + \Sigma & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right), \tag{10}$$

where

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

Nonstationary neural network kernel-based GP is an effective modeling option for modeling discontinuous data of varying sparseness.

are the output elevation values of the selected training data from the individual data sets,

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

are the input location values of the selected training data from the respective individual data sets, and

$$\Sigma = \mathrm{diag}\left[\sigma^2_{1\{1\ldots N_1\}}, \sigma^2_{2\{1\ldots N_2\}}, \sigma^2_{n\{1\ldots N_{nt}\}}\right]$$

is the diagonal noise matrix, where each noise term is repeated as many times as the number of training data taken from the corresponding data set, denoted by $N_1, N_2, \ldots, N_{nt}$. The experiments performed in [31] used the NN kernel, although any kernel [7] may be used so long as the same kernel is used for modeling each of the individual data sets. 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. $K(X, X)$, $K(X_*, X_*)$, and $K(X_*, X)$ can be similarly defined.

$$\bar{f}_* = K(X_*, X)[K(X, X) + \Sigma]^{-1}\mathbf{z}. \tag{11}$$

$$\mathrm{cov}(f_*) = K(X_*, X_*) - K(X_*, X)[K(X, X) + \Sigma]^{-1}K(X, X_*). \tag{12}$$

Conditioning (10) on f_* given $\mathbf{z}$ yields (11) and (12). The mean and variance of the elevation estimate can, thus, be obtained by applying (11) and (12), incorporating multiple data sets in the component terms as shown earlier. This estimate is the conditional estimate at a desired point, given the multiple and possibly multimodal data sets and their respective GP models. The data sets may, thus, be fused to generate integrated and comprehensive terrain models.

Learning of hyperparameters is based on the maximum marginal-likelihood framework demonstrated in [11]. Only one set of hyperparameters are used with the noise parameter alone varying across data sets. These hyperparameters may be obtained by selecting training data from each of the data sets and doing a joint learning exercise. This method is computationally expensive and is limited by the computational resources available but can be done if the learning approximation described in the "Approximation During GP Learning" section is used.

An alternative approach to the GP-hyperparameter learning is used in [31] that is based on constrained optimization. Here, the first data set (any one) in consideration is modeled using standard GP modeling procedure described in [11]. Thereafter, for each other data set, a constrained optimization method is adopted where the hyperparameters obtained before are reused, the noise hyperparameter alone being modified to model the successive data sets. This approach is akin to modeling successive data sets in terms of the existing data set and then fusing them using (11) and (12). The advantages of this approach include the bounding of the computational complexity and the ability to run the optimization operations in parallel. The former is particularly important in the context of this article.

Equations (11) and (12) provide the mean elevation estimate and uncertainty (covariance) of GP sampled at points X_*, assuming no noise in the query points. If the predictions need to be made at query points that are as uncertain as the data at hand, (12) is modified as:

$$\mathrm{cov}(f_*) = K(X_*, X_*) + R(X_*) - K(X_*, X)[K(X, X) + \Sigma]^{-1}K(X, X_*), \tag{13}$$

where $R(X_*)$ represents the noise or uncertainty of the query points themselves. In the homoscedastic case (single data set modeling as in the "GP Terrain Modeling" section), this is typically taken to be equal to the noise variance learned from the data. In the heteroscedastic case, this is not known. For heteroscedastic GP regression, estimation of the noise hyperparameters of the data points and the query points is a key issue. Goldberg et al. [16] and Kersting et al. [19] deal with the problem by maintaining two GPs: 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 (11) and (13). The latter GP is the key issue, as it provides the noise values to the former GP. Both [16] and [19] 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. Because the query points can be assumed to be as noisy as the training data, and the local approximation methodology toward GP regression [11] is adopted, the query points are assigned a noise value, that is the expected value (weighted average) of the noise terms of data taken from the individual data sets. More details on this may be found in [31].

The article [31] and its corresponding technical report detail numerous experiments conducted. Data fusion was demonstrated by showing that the uncertainty at a given set of test points from a data set never increased when it is fused with one or more other data sets. The usefulness of data fusion was demonstrated when the resulting output (post fusion) had lower prediction errors than when they were considered individually. One sample outcome is shown here. Figure 3 depicts the three data sets overlaid on each other to clarify the overall picture of the terrain in consideration. The points in blue represent laser scan 1 [dense RIEGL (http://www.riegl.com/) laser scan spanning 2,146.6 m × 2,302.1 m × 464.3 m and comprising more than 850,000 points], the points in red represent the second laser scan (dense RIEGL laser scan having about 400,000 points spread over 1,416.6 m × 2,003.4 m × 497.8 m), and finally, the points in green represent the GPS survey data (sparse GPS Survey having only about 34,530 points spread over 1,437.2 m × 1,879.5 m × 380.5 m).

Figures 4 and 5 depict the surface map and uncertainty estimates obtained after fusing the GPS data with the two laser scanner data sets.

Mathematical Properties

Equations (11) and (12) provide the batch fusion estimator, i.e., they provide the conditional mean and variance in elevation given all the data sets taken together.

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 detailed derivation is not included here because of the paucity of space; however, it is based on the following idea—without loss of generality, the difference between the posterior uncertainty using a single data set to that obtained using two data sets can be shown to be a positive semidefinite matrix. This change in uncertainty will be based on the information gain provided by points of the successive data sets to the prediction at a query point using the first data set. For two data sets, this is specified by:

$$\alpha_{21} = \begin{bmatrix} K_{11}^{-1} K_{12} \tilde{K}^{-1} K_{21} K_{11}^{-1} & -K_{11}^{-1} K_{12} \tilde{K}^{-1} \\ -\tilde{K}^{-1} K_{21} K_{11}^{-1} & \tilde{K}^{-1} \end{bmatrix}, \qquad (14)$$

where $\tilde{K} = K_{22} - K_{21} K_{11}^{-1} K_{12}$, $K_{11} = K(X_1, X_1) + \sigma_1^2 I$ represents the covariance matrix of the training data selected (for a query point) from the first data set. $K_{12} = K(X_1, X_2)$, $K_{21} = K(X_2, X_1)$, and $K_{22} = K(X_2, X_2) + \sigma_2^2 I$ can be similarly defined. This term is always positive semidefinite, guaranteeing that the uncertainty will either remain the same or decrease but never increase.

The information gain term (14) can be used to derive the conditional mean and variance in a recursive form as shown in [31]. This form enables a recursive fusion process wherein the previous best estimate (and its uncertainty) together with the information gain from the new data can be used to derive the new fused elevation estimate and its uncertainty. The article [31] provides more details on all aspects of this approach.

DGPs for Data Fusion

MOGPs (or multitask GPs) or DGPs extend the GP approach outlined before to handle multiple correlated outputs

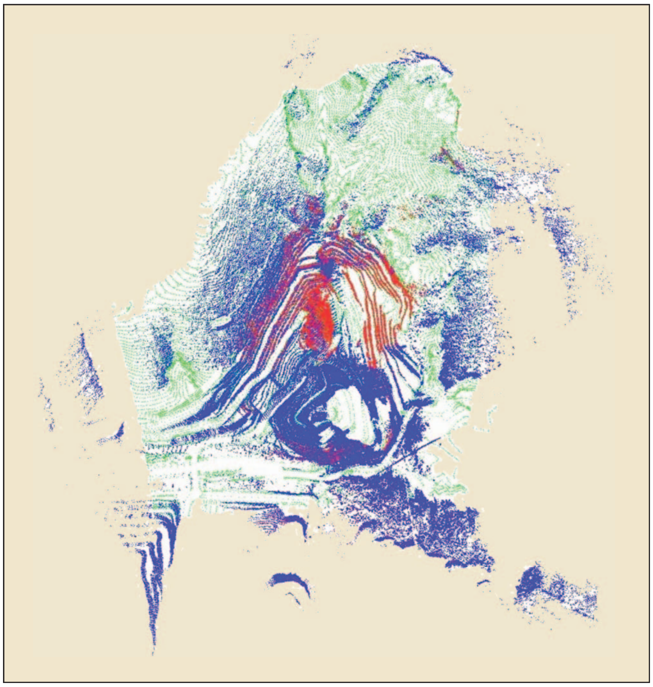

Figure 3. *The three (GPS survey and two laser scans) data sets overlaid on one another for a clearer picture of the site in consideration.*

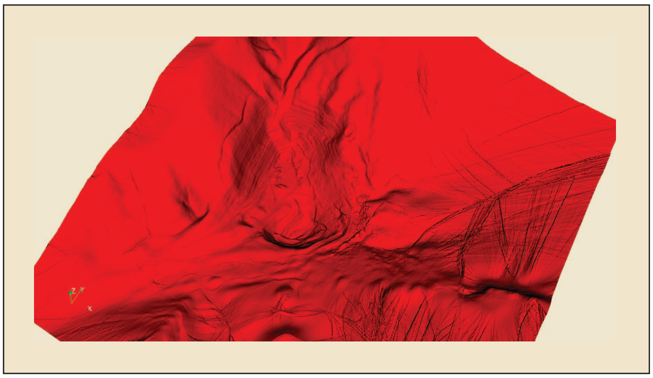

Figure 4. *Output of GP fusion algorithm applied to the Tom Price data sets (GPS data and the two laser scanner data sets). The test data comprises 1 million points. The surface map of the output elevation map is depicted in the image.*

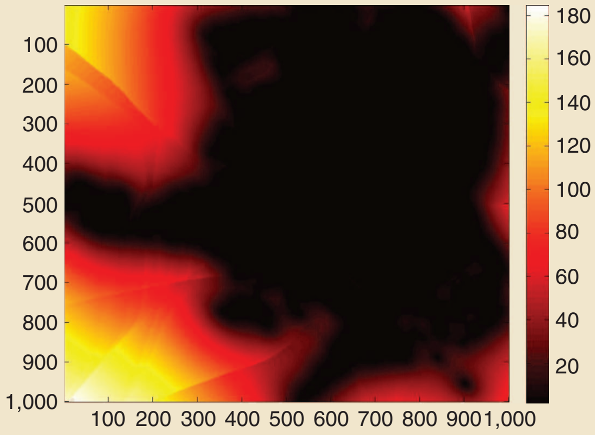

Figure 5. *Uncertainty (in meters) of the predicted elevation map obtained from the GP fusion of the GPS data and the two laser scanner data sets. The image corresponds to a top–down view as shown in Figure 3. Fringe areas that are not well supported by the individual data sets observe high-prediction uncertainty.*

Heteroscedastic GP and dependent GP concepts are applied to perform data fusion.

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. Two works in this area that have inspired this approach to data fusion include [20] and [21]. In [20], the shared covariance function is learned as a product of individual covariance functions and an intertask similarity matrix. Boyle and Frean [21] use the process convolution approach [22] to derive closed-form solutions to auto- and cross-covariance functions for two DGPs. The approach presented in this section integrates both of these ideas to allow for increased flexibility in learning DGP models. It has been presented in detail for stationary (SQEXP) and nonstationary (NN) kernels in [32] and [33], respectively.

The objective is to model terrain data obtained as (x, y, z) coordinates from multiple and multimodal data sets. Given the GP models of these data sets (as obtained earlier), the objective then would be to estimate an elevation map at any chosen resolution and any chosen region of the terrain under consideration. This can be achieved by performing a conditional estimation given the different data sets/their GP models. In the context of GPs, this amounts to GP regression. The problem can be specified as shown in (9). This estimation will need to take into account both the spatial correlation within each data set and the spatial correlation across data sets. Correlations between GPs can be modeled using autocovariances and cross-covariances between them. By performing GP regression that takes this information into account, conditional estimation can be achieved, which results in a fused elevation estimate, given the individual data sets.

The process convolution approach [22] is a generic methodology that 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 GPs this way is that it readily allows the GP to be extended to model more complex scenarios, one such scenario being the multioutput GPs or DGPs. The following formulation is based on [22] and [21].

Given that one 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) produces different data sets. The result of this convolution is denoted by $U_i(s)$. The observed data are 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. The mathematical formulation of the process convolution approach is given in (16).

$$Y_i(s) = U_i(s) + W_i(s). \tag{15}$$

$$U_i(s) = \int_s k_i(s - \lambda) \star X(\lambda) d\lambda. \tag{16}$$

When expressed using the process convolution approach, the auto- and cross-covariances between GPs, required for GP fusion, can be computed through a convolution integral as the kernel correlation, as demonstrated in [21]. In general, the main challenge in using this concept is the nontrivial derivation of closed-form solutions for auto- and cross-covariances for a given kernel. Vasudevan et al.'s study [32] is based on the SQEXP kernel and is inspired by [21]. The authors subsequently developed and applied this formalism for the nonstationary NN kernel in [33].

For two SQEXP GPs $N(0, k_i)$ and $N(0, k_j)$ with length-scale matrices Σ_i and Σ_j, respectively, the auto- and cross-covariances are specified by (17).

$$K_{ij}^U(x, x') = K_f * |\Sigma_i + \Sigma_j|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x - x')^T \Sigma_{ij} (x - x')\right), \tag{17}$$

where $\Sigma_{ij} = \Sigma_i(\Sigma_i + \Sigma_j)^{-1}\Sigma_j = \Sigma_j(\Sigma_i + \Sigma_j)^{-1}\Sigma_i$. K_{ii}^U represents the autocovariance of the ith data set with itself, and K_{ij}^U represents the cross-covariance between the ith and jth data sets, without considering the noise components of the data sets. The K_f term in (17) is inspired from [20]. This term models the task similarity between individual tasks. Incorporating it in the auto- and cross-covariances provides additional flexibility to the DGP modeling process. It is a symmetric matrix of size $nt * nt$ and is learned along with the other GP hyperparameters. For two NN GPs (following a similar notation), the auto- and cross-covariances may be specified by (18).

$$K_{ij}^U(x, x') = K_f * 2^{\frac{1}{2}} |\Sigma_i|^{\frac{1}{4}} |\Sigma_i + \Sigma_j|^{-\frac{1}{2}} |\Sigma_j|^{\frac{1}{4}} k(\mathbf{x}, \mathbf{x}', \Sigma_{ij}) \tag{18}$$

where Σ_{ij} is obtained as $\Sigma_{ij} = 2\Sigma_i(\Sigma_i + \Sigma_j)^{-1}\Sigma_j$, and 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 (5).

The covariance matrix term $K(X, X)$ in (7) and (8) is then specified as:

$$K = \begin{bmatrix} K_{11}^Y & K_{12}^Y & \dots & K_{1nt}^Y \\ K_{21}^Y & \dots & \dots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ K_{nt1}^Y & \dots & \dots & K_{nt\,nt}^Y \end{bmatrix}, \tag{19}$$

where

$$K_{ii}^Y = K_{ii}^U + \sigma_i^2 I, \tag{20}$$

$$K_{ij}^Y = K_{ij}^U. \tag{21}$$

K_{ii}^Y represents the autocovariance of the ith data set with itself, and K_{ij}^Y represents the cross-covariance between the ith and jth data sets. They also take the noise components of the data sets into consideration and are obtained as in (20) and (21), respectively. $K(X_*, X)$ denotes the covariance between the test data and the sets of input data (from the individual data sets) that are used for GP regression. It is given by:

$$K(X_*, X) = \left[K_{i1}^U(X_*, X_1), K_{i1}^U(X_*, X_2), \dots K_{i\,nt}^U(X_*, X_{nt})\right], \quad (22)$$

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 and is specified by:

$$K(X_*, X_*) = K_{ii}^U(X_*, X_*) + \sigma_i^2. \quad (23)$$

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

$$\mathbf{z} = \left[\mathbf{z_1}, \mathbf{z_2}, \dots, \mathbf{z_{nt}}\right]. \quad (24)$$

The hyperparameters of the system that need to be learned include $nt * (nt + 1)/2$ task similarity values, $nt * 2$ length-scale values of the individual kernels, and nt noise values corresponding to the noise in the observed data sets. GP learning is performed by the maximum marginal-likelihood estimation process [11], [32], [33] using the approximations introduced earlier. In the context of modeling a single terrain using multiple and multimodal data sets, for each point, the GP that is spatially closest to the test point is chosen for performing GP regression. The regression takes into account spatial correlation with other data sets as described.

The reports [32] and [33] describe the approach in detail, provide derivations for the math, and present preliminary results demonstrating data fusion (for multiple, multimodal data sets), as well as simultaneous elevation and color modeling. Further experimentation and optimization is intended to improve results and perform comparison/benchmarking experiments. One sample result from [32] is showcased in Figures 6–8. In Figure 7, about 2,550 points were used for training each task (elevation, red, green, and blue). The average mean squared error (MSE) between GP prediction and ground truth values obtained over the 100,000 points were 0.0524 m^2 for elevation and 0.0131, 0.0141, and 0.0101 squared units for red, green, and blue, respectively. The output demonstrates the ability of the approach to simultaneously handle elevation and color data in the context of terrain modeling. Figure 8(a) shows the output when the entire data set (Figure 6) is fused with patch 2 of the same data set—this represents a case of data fusion between two overlapping data sets. Figure 8(b) shows the output when two nonoverlapping patches of the original data set, patches 1 and 2 of Figure 6, are fused to predict the model for the entire data set. As shown in Figure 6, even in the area where no data is observed (patch 3), the DGP formalism produces a best estimate given the data sets at hand. These estimates were verified to be within two standard deviations ($2\,\sigma$) of the ground truth—see [32] for a figure showing this. In both the figures, the red and green points are the training points used from each data set, and the blue ones are generated from the DGP model.

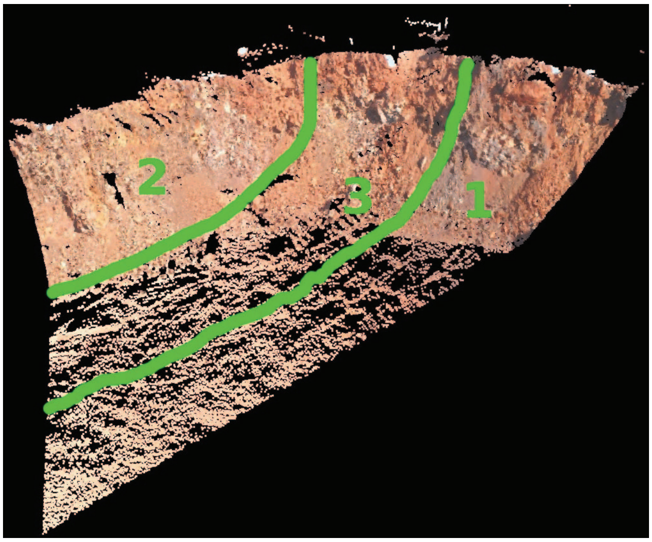

Figure 6. *Small section of a single RIEGL laser scan from Mt. Tom Price, Australia. The data set has about 150,000 points with both elevation and color (RGB) data. For the purpose of the data fusion experiments (see Figure 8), the data set is split into three patches as shown in the figure.*

Conclusions

This article reviewed recent efforts in applying a machine learning approach (GPs and models derived from them) toward solving a complex, real-world problem (large-scale terrain modeling) with significant practical applications (mining automation being one example). Specifically, this article addressed the problems of modeling and data fusion using GPs applied to large-scale problems. On the modeling front, it showed that a single NN-based GP was powerful enough to be able to successfully model complex terrain data, taking into account the

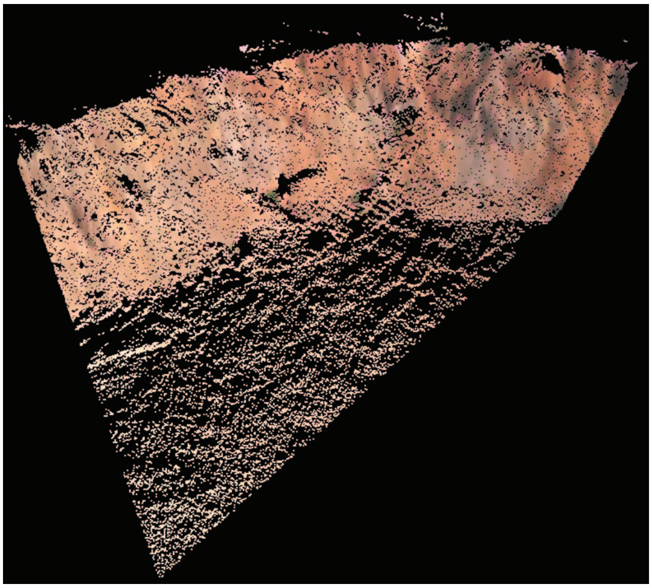

Figure 7. *A SQEXP kernel-based MOGP being used to simultaneously model and predict elevation and color (RGB) data at 100,000 test points taken from the Tom Price data set (see Figure 6).*

GP approximation methods are applied in both learning and inference enabling the method to handle data sets having millions of points.

local structure and preserving much of the spatial features in the terrain. To ensure scalability of the proposed methods, approximations were introduced in both learning and inference stages. Two approaches based on heteroscedastic GPs and DGPs were proposed to address a relatively unexplored problem—data fusion using GPs. The latter approach being more generic is shown to cope with nonoverlapping data and simultaneously model multiple properties of terrain data. Key distinguishing characteristics of this study include the use of the nonstationary NN kernel, the approximation methods used, the development of the nonstationary DGP approach, as well as the scale of the data used and the extent of experimentation. Further experimentation, comparison, and benchmarking are currently being undertaken for the data fusion approaches to fully understand them.

The study presented here provides a means to

1) model large-scale sensory data acquired at different degrees of sparseness
2) provide a nonparametric, multiresolution, and probabilistic representation of the large-scale terrain
3) provide an unbiased estimator for the data at any desired resolution
4) handle sensor incompleteness by providing an unbiased estimates of data that are unavailable (because of occlusions for instance); no assumptions on the structure of the data are made, and a best prediction is made using only the data at hand
5) handle sensor uncertainty by modeling the uncertainty in data in a statistically sound manner
6) handle spatial correlation between data
7) integrate (data fusion) multiple, multimodal data sets of a common underlying terrain.

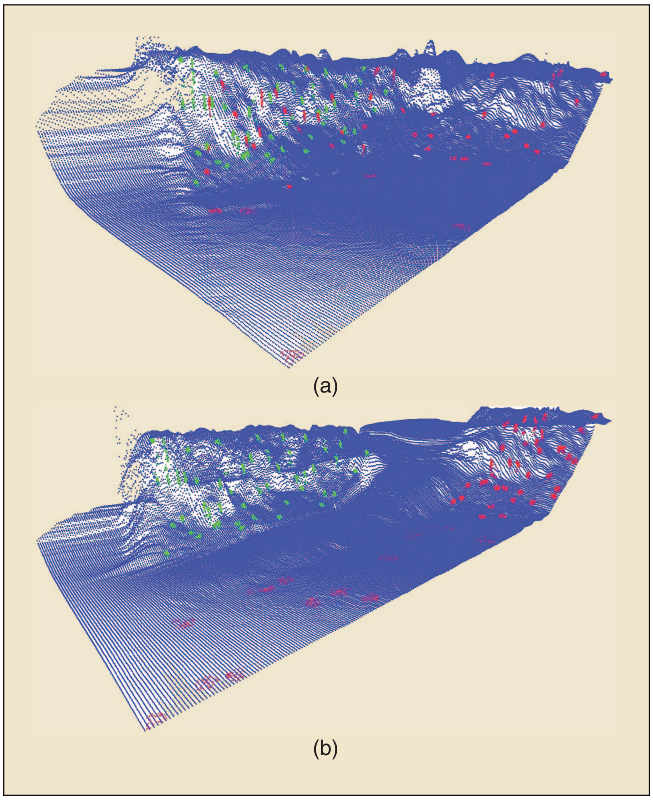

Figure 8. Output of using the DGP formalism with the SQEXP kernel to perform data fusion. (a) Fusing single-sensor overlapping data sets. (b) Fusing single-sensor nonoverlapping data sets.

Acknowledgments

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

Keywords

Gaussian process, terrain modeling, mapping, sensor fusion, field robotics, mining robotics.

References

[1] S. Lacroix, A. Mallet, D. Bonnafous, G. Bauzil, S. Fleury, M. Herrb, and R. Chatila, "Autonomous rover navigation on unknown terrains: Functions and integration," *Int. J. Robot. Res.*, vol. 21, no. 10–11, pp. 917–942, 2002.

[2] R. Triebel, P. Pfaff, and W. Burgard, "Multi-level surface maps for outdoor terrain mapping and loop closing," in *Proc. Int. Conf. Intelligent Robots and Systems (IROS)*, Beijing, China, Oct. 2006, pp. 2276–2282.

[3] J. Leal, S. Scheding, and G. Dissanayake, "3D mapping: A stochastic approach," in *Proc. Australian Conf. Robotics and Automation*, Nov. 2001, pp. 135–140.

[4] I. Rekleitis, J. Bedwani, D. Gingras, and E. Dupuis, "Experimental results for over-the-horizon planetary exploration using a LIDAR sensor," in *Proc. 11th Int. Symp. Experimental Robotics*, July 2008, pp. 65–78.

[5] H. Durrant-Whyte, "A critical review of the state-of-the-art in autonomous land vehicle systems and technology," Sandia Nat. Lab., Albuquerque, NM, Tech. Rep. SAND2001-3685, Nov. 2001.

[6] I. D. Moore, R. B. Grayson, and A. R. Ladson, "Digital terrain modelling: A review of hydrological, geomorphological, and biological applications," *Hydrol. Process.*, vol. 5, no. 1, pp. 3–30, 1991.

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

[8] P. K. Kitanidis, *Introduction to Geostatistics: Applications in Hydrogeology*. Cambridge, U.K.: Cambridge Univ. Press, 1997.

[9] G. Matheron, "Principles of geostatistics," *Econ. Geol.*, vol. 58, no. 8, pp. 1246–1266, 1963.

[10] 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," *J. Field Robot.*, vol. 26, no. 10, pp. 789–811, 2009.

[11] S. Vasudevan, F. T. Ramos, E. Nettleton, and H. Durrant-Whyte, "Gaussian process modeling of large-scale terrain," *J. Field Robot.*, vol. 26, no. 10, pp. 812–840, 2009.

[12] C. J. Paciorek and M. J. Schervish, "Nonstationary covariance functions for Gaussian process regression," in *Advances in Neural Information Processing*

Systems (NIPS) 16, S. Thrun, L. Saul, and B. Schölkopf, Eds. Cambridge, MA: MIT Press, 2004, pp. 273–280.

[13] R. Hadsell, J. Bagnell, and M. Hebert, "Accurate offroad terrain estimation with space carving kernels," in *Proc. Robotics Science and Systems (RSS)*, 2009.

[14] M. El-Beltagy and W. Wright, "Gaussian processes for model fusion," in *Proc. Int. Conf. Artificial Neural Networks (ICANN)*, 2001, pp. 376–383.

[15] R. Murray-Smith and B. Pearlmutter, "Transformations of Gaussian process priors," in *Deterministic and Statistical Methods in Machine Learning* (Lecture Notes in Artificial Intelligence 3635), J. Winkler, M. Niranjan, and N. Lawrence, Eds. New York: Springer-Verlag, 2005, pp. 110–123.

[16] P. W. Goldberg, C. K. I. Williams, and C. M. Bishop, "Regression with input-dependent noise: A Gaussian process treatment," in *Advances in Neural Information Processing Systems (NIPS) 10*, M. I. Jordan, M. J. Kearns, S. A. Solla, and L. Erlbaum, Eds. Cambridge, MA: MIT Press, 1998, pp. 493–499.

[17] M. Yuan and G. Wabha, "Doubly penalized likelihood estimator in heteroscedastic regression," *Stat. Probab. Lett.*, vol. 69, no. 1, pp. 11–20, Aug. 2004.

[18] Q. Le, A. Smola, and S. Canu, "Heteroscedastic Gaussian process regression," in *Proc. Int. Conf. Machine Learning (ICML)*, 2005, pp. 489–496.

[19] K. Kersting, C. Plagemann, P. Pfaff, and W. Burgard, "Most likely hetereoscedastic Gaussian process regression," in *Proc. Int. Conf. Machine Learning (ICML)*, 2007, pp. 393–400.

[20] 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.

[21] 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.

[22] D. Higdon, "Space and space-time modeling using process convolutions," in *Quantitative Methods for Current Environmental Issues*, C. W. Anderson, V. Barnett, P. C. Chatwin, and A. H. El-Shaarawi, Eds. New York: Springer-Verlag, 2002, pp. 37–54.

[23] H. Wackernagel, *Multivariate Geostatistics: An Introduction with Applications*. New York: Springer-Verlag, 2003.

[24] R. M. Neal, *Bayesian Learning for Neural Networks* (Lecture Notes in Statistics 118). New York: Springer-Verlag, 1996.

[25] C. K. I. Williams, "Computation with infinite neural networks," *Neural Comput.*, vol. 10, no. 5, pp. 1203–1216, 1998.

[26] C. K. Williams, "Prediction with Gaussian processes: From linear regression to linear prediction and beyond," in *Learning in Graphical Models*, M. Jordan, Ed. New York: Springer-Verlag, 1998, pp. 599–622.

[27] K. Hornik, "Some new results on neural network approximation," *Neural Netw.*, vol. 6, no. 8, pp. 1069–1072, 1993.

[28] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*. New York: Springer-Verlag, 1993.

[29] S. O'Callaghan, F. Ramos, and H. Durrant-Whyte, "Gaussian process occupancy maps incorporating sensor and location uncertainty," in *Proc. Int. Conf. Robotics and Automation (ICRA)*, 2010.

[30] J. Quinonero-Candela, C. E. Rasmussen, and C. K. I. Williams, "Approximation methods for Gaussian process regression," in *Large-Scale Kernel Machines*, L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, Eds. Cambridge, MA: MIT Press, 2007, pp. 203–223.

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

[32] S. Vasudevan, F. T. Ramos, E. Nettleton, and H. Durrant-Whyte, "Dependent Gaussian processes for data fusion in large-scale terrain modeling," Australian Centre for Field Robot., The Univ. Sydney, Tech. Rep. CMA003.109, 2010.

[33] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Nonstationary dependent Gaussian processes for data fusion in large-scale terrain modeling," Australian Centre for Field Robot., The Univ. Sydney, Tech. Rep. CMA003.110, 2010.

[34] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. 2nd ed. New York: Wiley, 2001.

Shrihari Vasudevan received the B.E. degree in computer science and engineering from the University of Madras, India, in 2002, the M.S. degree in computer science/intelligent robotics from the University of Southern California, in 2004, and the D.Sc. degree in computer science/robotics from the Swiss Federal Institute of Technology Zurich, Switzerland, in 2008. Since then, he has been a research fellow at the Australian Centre for Field Robotics, University of Sydney. His research interests include field and service robotics, especially perception, sensor fusion, learning, and representation methods toward the development of intelligent autonomous systems.

Fabio Ramos received the B.Sc. and the M.Sc. degrees in mechatronics engineering from the University of Sao Paulo, Brazil, in 2001 and 2003, respectively, and the Ph.D. degree from the University of Sydney, Australia, in 2007. Since 2007, he has been an Australian Research Council (ARC) research fellow at the Australian Centre for Field Robotics. He has over 60 peer-reviewed publications and received the Best Paper Award at the International Conference on Intelligent Robots and Systems (IROS). He is an associate editor for ICRA and IROS, and a program committee member for RSS, AAAI, and IJCAI. His research interests include statistical learning techniques for large-scale regression and classification problems, stochastic spatial modeling, and multisensor perception with applications in robotics and mining.

Eric Nettleton received a B.E. degree in mechatronic engineering from the University of Sydney in 1997. From 1998 to 1999, he worked as a technical engineer on robotics programs at the University of Sydney's Australian Centre for Field Robotics, before completing a Ph.D. degree in field robotics (1999 -2003) at the same institution. From 2003 to 2007, he worked for BAE Systems' corporate research facility, the Advanced Technology Centre in the United Kingdom. In 2007, he returned to the University of Sydney to lead the Rio Tinto Centre for Mine Automation. His research interests include data fusion, perception, and autonomous systems.

Hugh Durrant-Whyte received the B.Sc. degree in nuclear engineering from the University of London, United Kingdom, in 1983, and the M.S.E. and Ph.D. degrees, both in systems engineering, from the University of Pennsylvania in 1985 and 1986, respectively. From 1987 to 1995, he was a senior lecturer in engineering science at the University of Oxford, United Kingdom, and a fellow of Oriel College Oxford. From 1995 to 2002, he was a professor of mechatronic engineering at the University of Sydney. In 2002, he was awarded an Australian Research Council (ARC) Federation Fellowship. He also now leads the ARC Centre of Excellence in Autonomous Systems. His research interests include autonomous vehicle navigation and decentralized data fusion methods.

Address for Correspondence: *Shrihari Vasudevan, Australian Centre for Field Robotics, The University of Sydney, Darlington campus, Rose St. Building J04, N.S.W. 2006, Australia. E-mail: shrihari.vasudevan@ieee.org.*