Evaluation of Gaussian Processes for Large Scale Terrain Modeling

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

Abstract

This paper addresses the problem of large scale terrain modeling for a mobile robot. Building a model of large scale terrain that can adequately handle uncertainty and incompleteness in a statistically sound way is a challenging problem. A recent work [Vasudevan *et al.*, 2009] proposed non-stationary Gaussian processes (GP's) based on the neural network kernel as a solution to the problem. GP's naturally provide a multi-resolution representation of space, incorporate and handle uncertainty aptly and cope with incompleteness of sensory information. GP regression techniques may be applied to estimate and interpolate (to fill gaps in occluded areas) elevation information across the field. This paper presents results evaluating GP's for the problem of large scale terrain modeling. Extensive cross validation experiments were conducted on real world datasets obtained from different mine sites. These experiments are used to report statistically representative results that characterize the performance of the GP method for large scale and complex terrain modeling. They also compare its performance with grid based representations using many different interpolation techniques as well as triangulated irregular networks (TIN's); these represent the state-of-the-art in large scale terrain modeling.

1 Introduction and Related Work

Large scale terrain mapping is a difficult problem with wide-ranging applications in robotics, from space exploration to mining and more. For autonomous robots to function in such high-value applications, an efficient, flexible and high-fidelity representation of space is critical. The key challenges posed by this problem are that of dealing with the problems of uncertainty, incompleteness and handling 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 the significant scale of many domains.

State-of-the-art terrain 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, triangulated irregular networks (TIN's), contour models and their variants or combinations ([Durrant-Whyte, 2001] and [Moore *et al.*, 1991]). Each of them have their own strengths and preferred application domains. The former two are more popular in robotics. The latter represents the terrain as a succession of "isolines" of specific elevation (from minimum to maximum). They are particularly suited for modeling hydrological phenomena but otherwise offer no particular computational advantages in the context of this paper. Each of them, in their native form, do not handle spatially correlated data effectively.

Grid based methods represent space in terms of elevation data corresponding to each cell of a regularly spaced grid structure. The outcome is a 2.5D representation of space. The main advantage of this representation is simplicity. Limitations include the inability to handle abrupt changes, the dependence on grid size and the issue of scalability in large environments. In robotics, grid based methods have been exemplified by numerous works such as [Krotkov and Hoffman, 1994], [Ye and Borenstein, 2003], [Lacroix *et al.*, 2002] and more recently [Triebel *et al.*, 2006]. The main weaknesses observed in grid based representations are the lack of a statistically direct way of incorporating and managing uncertainty and the inability to appropriately handle spatial correlation.

Triangulated Irregular Networks (TIN's) usually sample a set of surface specific points that capture important aspects of the terrain surface to be modeled - bumps/peaks, troughs, breaks for example. The representation typically takes the form of an irregular network

of elevation points with each point linked to its immediate neighbors. This set of points is represented as a triangulated surface. TIN's are able to more easily capture sudden elevation changes and are also more flexible and efficient than grid maps in relatively flat areas. In robotics, TIN's have been used extensively, see [Leal *et al.*, 2001] and [Rekleitis *et al.*, 2008]. TIN's may be efficient from a survey perspective as few points are hand-picked. However [Triebel *et al.*, 2006] points out that for dense sensor data, while they are accurate and can easily be textured, they have a huge memory requirement which grows linearly with the number of scans. The assumption of spatial statistical independence may render the method ineffective at handling incomplete data as the elemental facet of the TIN (planar triangle) may approximate a complicated surface. This however depends on the choice of the sensor and the data density obtained.

In order to appropriately handle the issues of uncertainty, incompleteness and handling spatially correlated data, [Vasudevan *et al.*, 2009] proposed a novel approach to modeling complex and large scale terrain using non-stationary Gaussian processes (GP's) based on the neural network kernel. The approach produced a probabilistic, multi-resolution, continuous domain and non-parametric representation of the terrain. The model could be used to generate 2.5D elevation maps at any desired resolution. The estimation and interpolation (to fill in voids in sensor data) required to produce these maps was done using Gaussian process regression techniques. The model was probabilistic in that it could account for data uncertainty in a statistically correct way and also produce an uncertainty estimate for each point of interest in the final elevation map produced. A novel local approximation method based on a "moving window" implemented using KD-Trees was used to enable the algorithm to scale to large scale datasets.

This paper presents a detailed experimental evaluation of the approach presented in [Vasudevan *et al.*, 2009]. It has three main objectives -

1. Evaluate the GP modeling technique proposed in [Vasudevan *et al.*, 2009] in a way that provides for statistically representative results.
2. Provide benchmarks to compare the proposed technique (interpolation component) with many of the standard interpolation techniques for grid/elevation maps as well as TIN's.
3. Understand the strengths and applicability of different techniques.

The contribution of this work is thus a more comprehensive and statistically representative understanding of the Gaussian process method for terrain modeling as well as a comparison of the GP approach with the state-of-the-art, namely grid maps and TIN's. Experiments conducted on large scale datasets obtained from real mining applications, including a sparse mine planning dataset that is representative of a GPS based survey as well as dense Laser scanner based surveys, clearly demonstrate the viability and effectiveness of the Gaussian process representation of large-scale terrain. The paper also compares the performance of stationary and non-stationary kernels in the context of GP modeling of large scale terrain.

2 GP modeling of terrain data

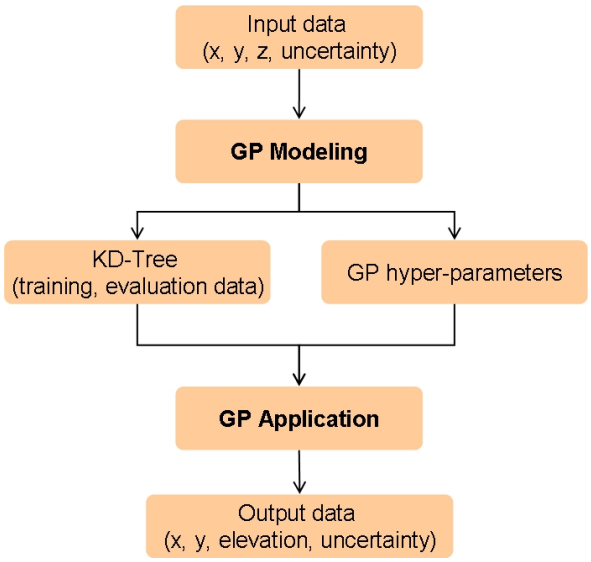


Figure 1: The overall process of learning and using Gaussian processes to model terrain: Data from any sensor along with noise estimates are provided as the input to the process. The modeling step then learns the appropriate Gaussian process model (hyper-parameters of the kernel) and the result of the modeling step are the GP hyperparameters together with the training/evaluation data. These are then used to perform GP regression to estimate the elevation data across any grid of points of the users interest. The application step produces a 2.5D elevation map of the terrain as well as an associated uncertainty for each point.

The overall process of using Gaussian processes to model large scale terrain is shown in Figure 1. The Gaussian process modeling method is described here in brief and the interested reader is directed to [Vasudevan *et al.*, 2009] for further information on all aspects of it.

Typically, for dense and large data sets, not all data is required for learning the GP model. Such an approach would not scale for reasons of computational complexity. Thus, a sampling step may need to be included. This sampling may be uniform or random and could also

use heuristic approaches such as preferential sampling from areas of high gradient. The dataset is sampled into three subsets - training, evaluation and testing. The training data is used to learn the GP model corresponding to the data. The training and evaluation data together are stored in the KD-Tree [Preparata and Shamos, 1993] for later use in the application or inference process. The KD-Tree provides for rapid data access in the inference process. A local approximation method based on a "moving window" or nearest-neighbor methodology is used to address scalability issues related to applying the GP modeling approach to large scale data sets. The approximation method results in a fixed computational complexity of $O(m^3)$, $m << N$ for GP regression at each test point, where m is the predefined number of nearest neighbors used and N is the set of all training and evaluation data stored in the KD-Tree. The testing data is the subset of data over which the GP model is evaluated. In this work, the mean squared error between the predicted and actual elevation, computed over the testing subset, is the performance metric used.

Gaussian processes [Rasmussen and Williams, 2006] provide a powerful framework for learning models of spatially correlated and uncertain data. 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 the problem at hand, 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 *squared-exponential* 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; $\Sigma = \begin{bmatrix} l_x & 0 \\ 0 & l_y \end{bmatrix}^{-2}$ is the length-scale matrix, a measure of how quickly the modeled function changes in the directions x and y; 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 *neural network* kernel ([Neal, 1996], [Williams, 1998a] and [Williams, 1998b]) is specified by

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

where β is a bias factor, Σ is the length scale matrix as described before and l_x , l_y , σ_f , β constitute the kernel hyperparameters.

The main difference between these two kernels is that the squared-exponential kernel, being a function of $|\mathbf{x} - \mathbf{x}'|$ is stationary (invariant to translation) whereas the neural network function is not so. In practice, the squared exponential function has a smoothing or averaging effect on the data. The neural network covariance function proves to be much more effective than the squared exponential covariance function in handling discontinuous (rapidly changing) data.

Learning or 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 squared-exponential kernel, this amounts to determining the values $\theta = \{l_x , l_y , \sigma_f , \sigma_n\}$ and for the neural network kernel, 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 problem in a Maximum Marginal Likelihood Estimation framework and subsequently solving a non-convex optimization problem.

Applying the GP model amounts 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 2.5D elevation map can then be used directly or as a surface map for various applications. This is achieved by performing Gaussian process regression at a set of query points, given the training/evaluation data sets and the GP kernel with the learnt hyperparameters.

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_* would also be similarly distributed. Applying this idea, the mean and variance of the elevation estimates at any desired test point can be specified by equations 6 and 7 respectively.

$$\bar{f}_* = k_*^T (K + \sigma_n^2 I)^{-1} \mathbf{z}, \tag{6}$$

and variance

$$V[f_*] = k(x_*, x_*) - k_*^T (K + \sigma_n^2 I)^{-1} k_* , \tag{7}$$

where $K = K(X, X)$ is the covariance matrix of the n training data and $k_* = K(X, x_*)$ is the covariance matrix between the training data and the test point at which the elevation must be estimated.

3 Approach

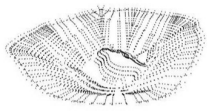
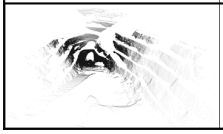
Image	Dataset / Type	Size / Resolution
	Tom Price (RIEGL Laser)	~ 1.8 million points (with RGB color) ~ 135 m x 72 x 13.8 m
	Kimberlite Mine (Mine planning data → GPS Survey)	4612 points ~ 2.2 x 2.3 x 0.25 km
	West Angelas (RIEGL Laser)	~ 500,000 points ~ 1.8 x 0.5 x 0.19 km

Figure 2: Datasets used for the experimental evaluation. The Tom Price and Kimberlite Mine datasets were used for evaluation in [Vasudevan *et al.*, 2009]. The West Angelas dataset is a rich new dataset that has been additionally used in this work for a more comprehensive evaluation of the GP modeling method.

The method adopted to achieve the statistical evaluation and benchmarking objectives of this paper (see section 1), was to perform extensive cross validation experiments on each of the datasets used in [Vasudevan *et al.*, 2009]. These datasets include the Tom Price and Kimberlite Mine datasets (see Figure 2). In addition, a third data set is also used to further test the GP modeling method. This also consisted of a laser scan but over a much larger area than the Tom Price data set. It comprises of about 534,460 points spread over 1.865 x 0.511 sq km. The maximum elevation of this dataset is about 190 m. This dataset is both rich in features (step like wall formations, voids of varying sizes) as well and large in scale; a large part of the dataset is composed of only sparse data. This dataset is referred to as the "West Angelas" dataset for the rest of this paper. A summary of the 3 datasets used is depicted in Figure 2.

The GP modeling using the squared exponential (SQEXP) and neural network (NN) kernels are compared with each other as well as with parametric, non-parametric and triangle based interpolation techniques. A complete list and description of the techniques used in this experiment is included below.

1. GP modeling using SQEXP and NN kernels. Details on the approach and the kernels can be found in Section 2 and [Vasudevan *et al.*, 2009].
2. Parametric Interpolation methods - These methods attempt to fit the given data to a polynomial surface of the chosen degree. The polynomial coefficients, once found, can be used to interpolate or extrapolate as required. The paramteric interpolation techniques used here include linear (plane fitting), quadratic and cubic.
   (a) Linear (plane fitting) - A polynomial of degree 1 (plane in 2D) is fit to the given data. The objective is to find the coefficients $\{a_i\}_{i=0}^{2}$ for the equation
   $$z = a_0 + a_1 x + a_2 y\,,$$
   that best describes the given data by minimizing the residual error.
   (b) Quadratic fitting: A polynomial of degree 2 is fit to the given data. The objective is to find the coefficients $\{a_i\}_{i=0}^{5}$ for the equation
   $$z = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2\,,$$
   that best describes the given data by minimizing the residual error.
   (c) Cubic fitting - A polynomial of degree 3 is fit to the given data. The objective is to find the coefficients $\{a_i\}_{i=0}^{9}$ for the equation
   $$z = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2 + a_6 x^3 + a_7 x^2 y + a_8 xy^2 + a_9 y^3,$$
   that best describes the given data by minimizing the residual error.
3. Non-parametric Interpolation methods - These techniques attempt to fit a smooth curve through the given data and do not attempt to characterize all of it using a parametric form, as was the case with the parametric methods. They include piecewise linear or cubic interpolation, biharmonic spline interpolation as well as techniques such as nearest neighbor interpolation and mean-of-neighborhood interpolation.
   (a) Piecewise Linear interpolation - In this method, the data is gridded (associated with a suitable mesh structure) and for any point of interest, the 4 points of its corresponding grid cell are determined. A bilinear interpolation [Kidner *et al.*, 1999] of these 4 points yields the estimate at the point of interest. This interpolation basically uses the form:
   $$z = a_0 + a_1 x + a_2 y + a_3 xy\,,$$

where the coefficients are expressed in terms of the coordinates of the grid cell. It is thus truly linear only if its behavior is considered keeping one variable/dimension fixed and observing the other. It is also called linear spline interpolation wherein $n+1$ points are connected by n splines such that (a) the resultant spline would constitute a polygon if the begin and end points were identical and (b) the splines were continuous at each data point.

(b) Piecewise Cubic interpolation - This method extends the piecewise linear method to cubic interpolation. It is also called cubic spline interpolation wherein $n+1$ points are connected by n splines such that (a) the splines are continuous at each data point and (b) the spline functions are twice continuous differentiable, meaning that both the first and second derivatives of the spline functions are continuous at the given data points. The spline function also minimizes the total curvature and thus produces the smoothest curve possible. It is akin to constraining an elastic strip to fit the given $n+1$ points.

(c) Biharmonic spline interpolation - The biharmonic spline interpolation is based on [Sandwell, 1987]. It places Green functions at each datapoint and finds coefficients for a linear combination of these functions to determine the surface. The coefficients are found by solving a linear system of equations. Both the slope and the values of the given data are used to determine the surface. The method is known to be relatively inefficient, possibly unstable but flexible (in the sense of the number of model parameters) in relation to the cubic spline method. In one or two dimensions, this is known to be equivalent to the cubic spline interpolation as it also minimizes the total curvature.

(d) Nearest-neighbor interpolation - This method uses the elevation estimate of the nearest training/evaluation data as the estimate of the point of interest.

(e) Mean-of-neighborhod interpolation - This method uses the average elevation of the set of training/evaluation data as the estimate of the point of interest.

4. Triangle based interpolation - All techniques mentioned thus far are applied on the exact same neighborhood of training and evaluation data that is used for GP regression for any desired point. In essense, thus far, the comparison can be likened to one of comparing GP's with standard elevation maps using any of the aforementioned techniques, applied over exactly the same neighborhoods. Two other interpolation techniques are also considered for comparison - Triangle based linear and cubic interpolation [Watson, 1992]. They are meant to compare the GP approach with a TIN based modeling/interpolation method.

   (a) Triangle based linear interpolation - In this method, a Delaunay triangulation is performed on the data provided and then the coordinates of the triangle corresponding to the point of interest (in Barycentric coordinates [Watson, 1992]) are subject to linear interpolation to estimate the desired elevation.

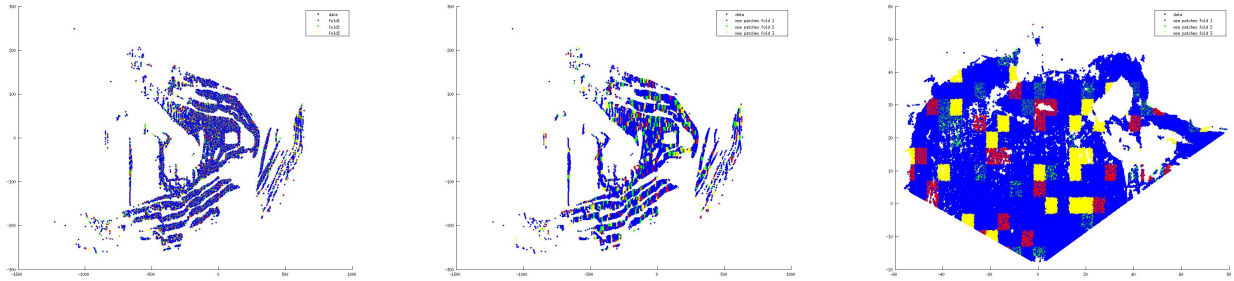
   (b) Triangle based cubic interpolation - In this method, a Delaunay triangulation is performed on the data provided and then the coordinates of the triangle corresponding to the point of interest (in Barycentric coordinates [Watson, 1992]) are subject to cubic interpolation to estimate the desired elevation.

   To make the comparison of GP's with TIN's more realistic, for these methods, the whole dataset was used to estimate the elevation at a point of interest.

Thus, this set of experiments will evaluate the proposed GP based approach, compare the GP model and its interpolation technique with a standard elevation map using many well known interpolation techniques and finally compare the proposed GP approach with a TIN based approach using linear or cubic interpolation.

4 Experiments

Taking inspiration from [Kohavi, 1995], a 10-fold cross validation was conducted using each dataset. Two kinds of cross validation techniques were considered - using uniform sampling to define the folds and using patch sampling to define the folds. In the uniform sampling method, each fold had approximately the same number of points and points were selected through a uniform sampling process - thus, the cross validation was also stratified in this sense. This was performed for all three datasets. Further to this, the patch sampling based cross validation was conducted for the two laser scanner datasets. In this case, the data was gridded into 5 m (for Tom Price dataset) or 10 m (for West Angelas dataset) patches. Subsequently a 10 fold cross validation was performed by randomly selecting different sets of patches for testing and training in each cycle. The uniform sampling method was meant to evaluate the method as such. The patch sampling method was done with a view of testing the different techniques for their robustness to handling



(a) Uniform sampling of West Angelas data

(b) Patch sampling of West Angelas data

(c) Patch sampling of Tom Price data

Figure 3: 3(a) depicts cross validation using uniform sampling. Points are uniformly sampled from the dataset to produce 10 equally sized folds and cross validation is performed by testing each fold while learning from the remaining folds. The figure depicts the data and test points from three folds. 3(b) and 3(c) depict cross Validation using patch sampling. The dataset is gridded into 5m (Tom Price) and 10m (West Angelas) patches. 10 fold cross validation is done by randomly selecting patches for training and testing in each cycle. The figures depict the test patches in three cycles of testing in the two laser scanner datasets.

occlusions or voids in data. Figure 3 depicts the two kinds of cross validation performed.

The results of the cross validation with uniform sampling experiments are summarized in Tables 1, 2 and 3. In each of these cases, only the interpolation technique of the GP using the SQEXP/NN kernels and the various standard interpolation methods are compared - ie. the test is applied after performing the fast local approximation step in the inference process. Thus, all the interpolation techniques are applied on exactly the same neighborhood of training/evaluation data. The tabulated results depict the mean and standard-deviation in MSE values obtained across the 10 folds of testing performed. The results of the cross validation with patch sampling experiments performed on the Tom Price and West Angelas datasets are respectively summarized in Tables 4 and 5.

5 Discussion

From the experimental results presented, many inferences on the applicability and usage of different techniques could be drawn. The results obtained validate the findings from [Vasudevan *et al.*, 2009]. They show that the NN kernel is much more effective at modeling terrain data than the SQEXP kernel. This is particularly true in relatively sparse and/or complex datasets such as the Kimberlite Mine or West Angelas datasets as shown in Tables 2, 3 and 5.

For relatively flat terrain such as the Tom Price scan data, the techniques compared performed more or less similarly. This is shown in Tables 1 and 4. From the GP modeling perspective, the choice of non-stationary or stationary kernel was less relevant in such a scenario. The high density of data in the scan was likely another contributive factor.

For sparse datasets such as the Kimberlite Mine dataset, GP-NN easily outperformed all the other interpolation methods. This is shown in Table 2. This is quite simply because most other techniques imposed apriori models on the data whereas the GP-NN was actually able to model and adapt to the data at hand much more effectively. Further, GP-SQEXP is not as effective as the GP-NN in handling sparse or complex data.

In the case of the West Angelas dataset, the data was complex in that it had a poorly defined structure and large areas of of the data were sparsely populated. This was due to large occlusions as the scan was a bottom-up scan from one end of the mine pit. However, the dataset spanned over a large area and local neighborhoods (small sections of data) were relatively flat. Thus, while the GP-NN clearly outperformed the GP-SQEXP, it performed comparably with the piecewise linear/cubic and triangle based linear/cubic methods and much better than the other techniques attempted. This is shown in the Tables 3 and 5. Thus, even in this case, the GP-NN proved to be a very competitive modeling option. GP-SQEXP was not able to handle sparse data or the complex structure (with an elevation of 190 m) effectively and hence performed poorly in relation to the aforementioned methods.

As evidenced by Tables 2, 3 and 5, the non-parametric and triangle based linear/cubic interpolation techniques performed much better than the parametric interpolation techniques, which fared poorly overall. This is ex-

Table 1: Tom Price Mine dataset 10-fold cross validation with uniform sampling. Comparison of Interpolation techniques.

	1000 test data per fold		10000 test data per fold	
Interpolation Method	Mean MSE (sq m)	Std. Dev. MSE (sq m)	Mean MSE (sq m)	Std. Dev. MSE (sq m)
GP Neural Network	0.0107	0.0012	0.0114	0.0004
GP Squared Exponential	0.0107	0.0013	0.0113	0.0004
Nonparametric Linear	0.0123	0.0047	0.0107	0.0013
Nonparametric Cubic	0.0137	0.0053	0.0120	0.0017
Nonparametric Biharmonic	0.0157	0.0065	0.0143	0.0019
Nonparametric Mean-of-neighborhood	0.0143	0.0010	0.0146	0.0007
Nonparametric Nearest-neighbor	0.0167	0.0066	0.0149	0.0017
Parametric Linear	0.0107	0.0013	0.0114	0.0005
Parametric Quadratic	0.0110	0.0018	0.0104	0.0005
Parametric Cubic	0.0103	0.0018	0.0103	0.0005
Triangulation Linear	0.0123	0.0046	0.0107	0.0013
Triangulation Cubic	0.0138	0.0053	0.0120	0.0017

Table 2: Kimberlite Mine dataset 10-fold cross validation with uniform sampling. Comparison of Interpolation techniques.

Interpolation Method	Mean MSE (sq m)	Std. Dev. MSE (sq m)
GP Neural Network	3.9290	0.3764
GP Squared Exponential	5.3278	0.3129
Nonparametric Linear	5.0788	0.6422
Nonparametric Cubic	5.1125	0.6464
Nonparametric Biharmonic	5.5265	0.5801
Nonparametric Mean-of-neighborhood	132.5097	2.9112
Nonparametric Nearest-neighbor	20.4962	2.5858
Parametric Linear	43.1529	2.2123
Parametric Quadratic	13.6047	0.9047
Parametric Cubic	10.2484	0.7282
Triangulation Linear	5.0540	0.6370
Triangulation Cubic	5.1091	0.6374

Table 3: West Angelas Mine dataset 10-fold cross validation with uniform sampling. Comparison of Interpolation techniques.

	1000 test data per fold		10000 test data per fold	
Interpolation Method	Mean MSE (sq m)	Std. Dev. MSE (sq m)	Mean MSE (sq m)	Std. Dev. MSE (sq m)
GP Neural Network	0.0166	0.0071	0.0219	0.0064
GP Squared Exponential	0.4438	1.0289	0.7485	0.7980
Nonparametric Linear	0.0159	0.0075	0.0155	0.0021
Nonparametric Cubic	0.0182	0.0079	0.0161	0.0021
Nonparametric Biharmonic	0.0584	0.0328	0.1085	0.1933
Nonparametric Mean-of-neighborhood	0.9897	0.4411	0.9158	0.0766
Nonparametric Nearest-neighbor	0.1576	0.0271	0.1233	0.0048
Parametric Linear	0.1019	0.0951	0.0927	0.0173
Parametric Quadratic	0.0458	0.0130	0.0390	0.0059
Parametric Cubic	0.0341	0.0109	0.0288	0.0038
Triangulation Linear	0.0162	0.0074	0.0157	0.0022
Triangulation Cubic	0.0185	0.0078	0.0166	0.0023

Table 4: Tom Price Mine dataset 10-fold cross validation with patch sampling. Comparison of Interpolation techniques.

	1000 test data per fold		10000 test data per fold	
Interpolation Method	Mean MSE (sq m)	Std. Dev. MSE (sq m)	Mean MSE (sq m)	Std. Dev. MSE (sq m)
GP Neural Network	0.0104	0.0029	0.0100	0.0021
GP Squared Exponential	0.0103	0.0029	0.0099	0.0021
Nonparametric Linear	0.0104	0.0047	0.0098	0.0040
Nonparametric Cubic	0.0114	0.0054	0.0108	0.0045
Nonparametric Biharmonic	0.0114	0.0046	0.0124	0.0047
Nonparametric Mean-of-neighborhood	0.0143	0.0036	0.0142	0.0026
Nonparametric Nearest-neighbor	0.0139	0.0074	0.0132	0.0048
Parametric Linear	0.0103	0.0029	0.0100	0.0021
Parametric Quadratic	0.0099	0.0029	0.0091	0.0022
Parametric Cubic	0.0097	0.0027	0.0605	0.1072
Triangulation Linear	0.0104	0.0047	0.0098	0.0039
Triangulation Cubic	0.0114	0.0054	0.0108	0.0045

Table 5: West Angelas Mine dataset 10-fold cross validation with patch sampling. Comparison of Interpolation techniques.

	1000 test data per fold		10000 test data per fold	
Interpolation Method	Mean MSE (sq m)	Std. Dev. MSE (sq m)	Mean MSE (sq m)	Std. Dev. MSE (sq m)
GP Neural Network	0.0286	0.0285	0.0291	0.0188
GP Squared Exponential	0.7224	1.1482	1.4258	2.0810
Nonparametric Linear	0.0210	0.0168	0.0236	0.0161
Nonparametric Cubic	0.0229	0.0175	0.0243	0.0165
Nonparametric Biharmonic	0.0555	0.0476	0.0963	0.0795
Nonparametric Mean-of-neighborhood	1.6339	0.8907	1.7463	1.2831
Nonparametric Nearest-neighbor	0.1867	0.0905	0.2248	0.1557
Parametric Linear	0.1400	0.0840	0.1553	0.1084
Parametric Quadratic	0.0612	0.0350	0.0658	0.0454
Parametric Cubic	0.0380	0.0196	0.0752	0.0880
Triangulation Linear	0.0212	0.0169	0.0235	0.0160
Triangulation Cubic	0.0232	0.0175	0.0245	0.0164

pected and intuitive as the parametric models impose apriori models over the whole neighborhood of data whereas the non-parametric and triangle based techniques basically split the data into simplexes (triangular or rectangular patches) and then operate only one such simplex for a point of interest. Thus their focus is more towards smooth local data fitting. The parametric techniques only performed well in the Tom Price dataset tests (Tables 1 and 4) due to the flat nature of the dataset. Among the non-parametric techniques, the linear and cubic also performed better than the others. The nearest-neighbor and mean-of-neighborhood methods were too simplistic to be able to handle any complex or sparse data.

The results prove that the GP-NN is a very versatile and competitive modeling option for a range of datasets, varying in complexity and sparseness. For dense datasets or relatively flat terrain (even locally flat terrain), the GP-NN will perform as well as the standard grid based methods employing any of the interpolation techniques compared or a TIN based representation employing triangle based linear or cubic interpolation. For sparse datasets or very bumpy terrain, the GP-NN will significantly outperform every other technique that has been tested in this experiment. When considering the other advantages of the GP approach such as Bayesian model fitting, ability to handle uncertainty and spatial correlation in a statistically sound manner, the GP approach to terrain representation seems to have a clear advantage over standard grid based or TIN based approaches.

Finally, note that in essence, these experiments compared the GP - Kriging interpolation technique with various other standard interpolation methods. The GP modeling approach can use any of the standard interpolation techniques through the use of suitably chosen/designed kernels. For instance, a GP-linear kernel performs the same interpolation as a linear interpolator, however, the GP also provides the corresponding uncertainty estimates among other things. The GP is more than just an interpolation technique. It is a Bayesian framework that brings together interpolation, linear least squares estimation, uncertainty representation, non-parametric continuous domain model fitting and the Occam's Razor principle (thus preventing the over-fitting of data). Individually, these are well known techniques; the GP provides a flexible (in the sense of using different kernels) technique to do all of them together.

6 Conclusion

This paper presented an extensive experimental evaluation of the Gaussian process terrain modeling approach presented in [Vasudevan *et al.*, 2009]. The modeling approach yielded a probabilistic, multi-resolution, continuous domain and non-parametric representation of large scale and complex terrain. The following conclusions could be drawn out of the experiments performed in this paper

1. The non-stationary neural network kernel based Gaussian process (GP-NN) proved to be a versatile and capable solution for the problem of modeling complex large scale terrain. This was demonstrated

by the ability of the GP-NN to be able to consistently perform well on different datasets of varying resolution, obtained from different sensory modalities and being representative of a range of scenarios.

2. The proposed Gaussian process modeling methodology will perform very competitively in comparison with grid maps or TIN's for dense datasets (such as that obtained from the RIEGL laser scanners). The latter two methods represent the state-of-the-art in both robotics research as well as in many industries. For sparse or complex (not flat) terrain, the proposed GP approach will significantly outperform grid maps using standard interpolation techniques or TIN's.

Acknowledgements

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

References

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

[Kidner *et al.*, 1999] D. Kidner, M. Dorey, and D. Smith. What's the point? Interpolation and extrapolation with a regular grid DEM. In *Fourth International Conference on GeoComputation*, Fredericksburg, VA, USA, 1999.

[Kohavi, 1995] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)*, volume 2 (12), page 1137:1143, 1995.

[Krotkov and Hoffman, 1994] E. Krotkov and R. Hoffman. Terrain mapping for a walking planatary rover. *IEEE Transactions on Robotics and Automation*, 10(6):728–739, 1994.

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

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

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

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

[Preparata and Shamos, 1993] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer, 1993.

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

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

[Sandwell, 1987] David T. Sandwell. Biharmonic spline interpolation of geos-3 and seasat altimeter data. *Geophysical Research Letters*, 14,2:139–142, 1987.

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

[Vasudevan *et al.*, 2009] Shrihari Vasudevan, Fabio Ramos, Eric Nettleton, and Hugh Durrant-Whyte. Gaussian process modeling of large scale terrain. In *the proceedings of the International Conference on Robotics and Automation (ICRA)*, 2009.

[Watson, 1992] David E. Watson. *Contouring: A Guide to the Analysis and Display of Spatial Data*. Pergamon (Elsevier Science, Inc.), 1992.

[Williams, 1998a] Christopher K. I. Williams. Computation with infinite neural networks. *Neural Computation*, 10(5):1203–1216, 1998.

[Williams, 1998b] Christopher K. I. Williams. Prediction with gaussian processes: From linear regression to linear prediction and beyond. In Michael Irwin Jordan, editor, *Learning in Graphical Models*, pages 599–622. Springer, 1998.

[Ye and Borenstein, 2003] C. Ye and J. Borenstein. A new terrain mapping method for mobile robots obstacle negotiation. In *UGV Technology Conference at the SPIE AeroSense Symposium*, volume 4, Orlando, FL, USA, April 2003.