Gaussian Processes for Multi-Sensor Environmental Monitoring

Philip Erickson, Michael Cline, Nishith Tirpankar, and Tom Henderson
School of Computing
University of Utah, Salt Lake City, UT, USA

***Abstract*— Efficiently monitoring environmental conditions across large indoor spaces (such as warehouses, factories or data centers) is an important problem with many applications. Deployment of a sensor network across the space can provide very precise readings at discrete locations. However, construction of a continuous model from this discrete sensor data is a challenge. The challenge is made harder by economic and logistical constraints that may limit the number of sensor motes in the network. The required model, therefore, must be able to interpolate sparse data and give accurate predictions at unsensed locations, as well as provide some notion of the uncertainty on those predictions. We propose a Gaussian process based model to answer both of these issues. We use Gaussian processes to model temperature and humidity distributions across an indoor space as functions of a 3-dimensional point. We study the model selection process and show that good results can be obtained, even with sparse sensor data. Deployment of a sensor network across an indoor lab provides real-world data that we use to construct an environmental model of the lab space. We seek to refine the model obtained from the initial deployment by using the uncertainty estimates provided by the Gaussian process methodology to modify sensor distribution such that each sensor is most advantageously placed. We explore multiple sensor placement techniques and experimentally validate a near-optimal criterion.**

I. INTRODUCTION

Consider the problem of monitoring various environmental conditions over an indoor space. In settings such as warehouses, factories or data centers, accurate estimates of environmental conditions (such as temperature, humidity, air particulate and composition, sound levels, static discharge, etc.) can be crucial. Modern computing technology provides both the hardware and software capabilities needed to solve such a monitoring problem. A single sensor mote may be capable of simultaneously monitoring many different conditions and logging data over time. Work has been done in developing such systems (see [1], [2]). There is no lack of commercial systems to gather such data. However, obtaining a spatially complete model from the data remains a challenge. Regression techniques address the hard problem of translating scattered sensor readings into a cohesive, continuous model. In the context of our problem, the indoor spaces to be monitored can be quite large (estimates in [3] put the average volume of a warehouse in the United States at over a million cubic feet). Driven by logistical and economic constraints, a feasible sensor network for monitoring the environment may be quite sparse, making the problem of accurate interpolation even more difficult. Determining the best sensor placement locations, so as to gather the most useful data, is therefore an important problem to address.

Dynamic alternatives to a fixed sensor network using mobile robots to take readings have been proposed (such as in [4]). This may provide a more scalable approach for monitoring large spaces, but the problems of interpolating the collected data and determining the ideal locations for the robots to visit remain. Though we focus on a static sensor network in this work, the results are directly applicable to environmental monitoring using mobile robots.

In response to these challenges, this paper proposes the use of a Gaussian process based approach to modeling the environment. The Gaussian process approach answers the above problems. First, it allows for effectively combining discrete sensor data in a continuous-domain function, allowing queries on the value of a variable to be made at any location. Second, the Gaussian process approach provides methods for calculating the uncertainty associated with a given estimate. This knowledge can be applied to determine ideal locations for taking sensor readings, which allows for achieving the desired precision using fewer resources.

The contribution of this paper is a novel approach to monitoring environmental variables across a 3-dimensional space. Specifically, we show that a Gaussian process model based on the *Matérn class* of covariance functions is able to successfully model the environment, outperforming the more commonly used *squared exponential* and *rational quadratic* covariance functions. We look at methods of optimizing sensor placement, extending these techniques to sensors that monitor multiple conditions, thereby finding the best location to gather all relevant data.

The object of our study is the temperature and humidity distribution over an indoor lab space. For initial investigation, we make use of realistic data sets obtained through simulation. For real-world experimentation, we deploy a sensor network across the lab to gather temperature and humidity data. In keeping with our aim at a solution that can scale, we limit our use of sensors while still obtaining meaningful, useful results.

II. GAUSSIAN PROCESSES

A Gaussian process is simply a collection of random variables which have a multivariate normal joint distribution. For any subset X consisting of k of these random variables, we can construct the vector $\boldsymbol{\mu}$ of their expectations and matrix $\boldsymbol{\Sigma}$ of the covariance between variables. Then the joint distribution of X is given by:

$$P(X = \boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^{k/2}|\boldsymbol{\Sigma}|}} \exp\Big(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\boldsymbol{T}}\boldsymbol{\Sigma}^{\boldsymbol{-1}}(\boldsymbol{x}-\boldsymbol{\mu})\Big)$$

We can index a single random variable by $\mathbf{x}$. The value then of a particular random variable $\mathbf{x}$ represents the value of some function f at the location $\mathbf{x}$. For example, a Gaussian process may consist of a collection of random variables that represent the function f, where $f(x, y)$ is the temperature at a given location in 2-dimensional space. If we index the random variables by $\mathbf{x_i} = (x_i, y_i)$, then $f(\mathbf{x_i})$ represents the temperature at location (x_i, y_i). The example is 2-dimensional, but Gaussian processes can be extended to higher dimensions. A Gaussian process, then, can be thought of as a distribution over functions. Given a mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$ for a Gaussian process, we can estimate the function it represents.

Gaussian processes are useful because they provide a powerful means of regression. In the context of our problem, this means effectively combining discrete sensor data into a continuous model, allowing us to estimate the true temperature and humidity functions at any location in our region of interest. Additionally, the Gaussian process model supplies us with information about the uncertainty of our predictions.

Constructing a Gaussian process model amounts to performing a few steps. First, some assumption about the covariance function (or *kernel*) must be made. The choice of covariance function reflects our assumptions about the characteristics of our data, so care must be taken in selection. Generally, the kernel is chosen from among a number of well-known functions. Free parameters in the covariance function (called *hyperparameters*) must be tuned to fit the particular data set. The process of choosing a kernel function and learning the hyperparameters from the observed training data is known as model selection. The selected model can then be used in the inference process: given a set of observations, inferring values at unobserved locations.

Details of this process as they relate to our problem are given in the following sections. For a more thorough explanation of the theory and various applications of Gaussian processes, the reader is referred to [5], from which these procedures are primarily drawn.

III. APPROACH

The purpose of this section is to detail the process we will follow in constructing and refining our environmental model. Our method for the initial deployment of the sensor network is explained. Details of the model selection process—the process of choosing a covariance function and learning the hyperparameters—are given. Using the learned Gaussian process to perform inference across the entire region—giving estimates about environmental conditions and the model's uncertainty on those predictions—is explained. Finally, we show how the model can be refined by using the information gained from the initial model to redistribute the sensor network in a more optimal way.

A. Initial Sensor Deployment

With no prior knowledge of the environment, initial deployment of the sensor network requires uninformed decisions about the placement of sensor motes. Our approach uses the Halton sequence [6] to generate 3-dimensional points evenly over the space. The Halton sequence is a quasi-random sequence—the point distribution tends to be uniform and avoids the clumping and gaps that generally occur in random placements. Figure 1 compares 100 randomly generated points (sampled from a uniform distribution) in the unit square with 100 points generated using the Halton sequence.

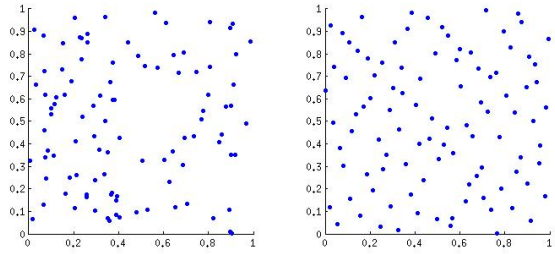

Fig. 1. Uniform random points (left) versus points from the Halton sequence (right).

We take this approach as a simple means of ensuring that our sensors have good coverage of the region. Sensor readings from this initial distribution serve as training data for the initial model selection process.

B. Covariance Functions

The process of model selection begins with choosing an appropriate covariance function. The covariance function encodes our assumptions about the structure of the underlying data, hence the choice is key to constructing a good Gaussian process model. Common choices are the squared exponential and the rational quadratic kernels. The Matérn kernel is suggested by [7], as the squared exponential may be too smooth for modeling many physical phenomena. We evaluate all three kernels for application in our environmental model. The covariance functions are shown in Table I. Note that here we have shown one particular case of the Matérn kernel.

TABLE I
COVARIANCE FUNCTIONS

Squared Exponential	$\sigma_f^2 \exp\left(-\frac{\mathbf{r^2}}{2\mathbf{L}^2}\right)$
Rational Quadratic	$\sigma_f^2 \left(1 + \frac{\mathbf{r^2}}{2\alpha\mathbf{L}^2}\right)^{-\alpha}$
Matérn	$\sigma_f^2 \left(1 + \frac{\sqrt{5}\mathbf{r}}{\mathbf{L}} + \frac{5\mathbf{r^2}}{3\mathbf{L^2}}\right) \exp\left(-\frac{\sqrt{5}\mathbf{r}}{\mathbf{L}}\right)$

For the given kernels, we define $\mathbf{r} = (\mathbf{x} - \mathbf{x}')$ with $\mathbf{x}$ and $\mathbf{x}'$ as the input variables. The matrix $\mathbf{L}$ contains the lengthscales for each of the different dimensions of $\mathbf{x}$, in our case:

$$\mathbf{L} = \begin{bmatrix} l_x & 0 & 0 \\ 0 & l_y & 0 \\ 0 & 0 & l_z \end{bmatrix}$$

The lengthscales, along with α and the output variance σ_f^2, constitute the hyperparameters for the different kernels. These hyperparameters need to be chosen to particularize the kernel to a given data set, as discussed in the next section.

In practice, modeling problems must account for some inherent error on the observations; however precise, a sensor will have some amount of noise. It is a common and reasonable assumption that the noise is additive Gaussian. In this case, we add a noise term σ_n^2 to the diagonal entries in the covariance matrix. This parameter can be left free and learned with the other hyperparameters, or can be constrained to a set value. Having a detailed characterization of our sensors, we constrain σ_n^2 to the known value.

C. Hyperparameters

Learning the hyperparameters for a Gaussian process particularizes the chosen covariance function to a given data set. We refer to this as training the Gaussian process. For the Matérn kernel, with σ_n^2 constrained, the vector $\theta = \{l_x, l_y, l_z, \sigma_f\}$ represents the hyperparameters. Learning the best hyperparameters is an optimization problem in the 4-dimensional hyperparameter space. The objective function we wish to maximize is the *log marginal likelihood*, given in equation 1.

$$\log(\mathbf{z}|X,\theta) = -\frac{1}{2}\mathbf{z}^T K_z^{-1}\mathbf{z} - \frac{1}{2}\log|K_z| - \frac{n}{2}\log(2\pi) \quad (1)$$

Here, X is the set of n training inputs—the set of locations at which sensor readings are taken. We define $\mathbf{z}$ as the set of training outputs—the set of temperature or humidity levels read at the sensor locations. Let $K(X,X)$ denote the $n \times n$ covariance matrix formed by taking the pairwise covariance between the elements in X. Then we define $K_z = K(X,X) + \sigma_n^2 I$, where I is the $n \times n$ identity matrix. Equation 1 gives us the likelihood of our observed readings, given the locations at which the readings were observed and the selected hyperparameters. Maximizing this likelihood gives the best set of hyperparameters for the chosen kernel. We perform this process for temperature and humidity data separately, resulting in a different set of hyperparameters for each variable.

Many possibilities exist for performing this optimization. Various gradient-based approaches have been proposed ([8], [9], [10]). We also evaluate a simulated annealing algorithm for maximizing the likelihood ([11]).

D. Applying the Initial Model

The model selection step results in a Gaussian process trained to the observed sensor readings. Let x_* index the random variable in our Gaussian process corresponding to the 3-dimensional point we wish to predict. As stated earlier, the value of the random variable represents the temperature or humidity at that point, the distribution of which is specified by some mean and covariance. Equations 2 and 3 specify the distribution of x_* conditioned on the training data. Let $\bar{f}_*$ and $V[f_*]$ denote the mean and variance of the distribution, respectively. Application of the model is done by evaluating these equations at every point of interest.

$$\bar{f}_* = k_*^T K_z^{-1}\mathbf{z} \quad (2)$$

$$V[f_*] = k(x_*, x_*) - k_*^T K_z^{-1} k_* \quad (3)$$

Here, we take k_* to be the vector of covariances between x_* and the training inputs and $k(x_*, x_*)$ to be the variance of x_*.

Work done in [12], [13], [14], [15] proposes the use of a KD-tree structure for a fast local approximation. For a training set of n observations, the inference process requires calculating and inverting an $n \times n$ covariance matrix, which gets computationally expensive as n gets large. For inference about x_*, the KD-tree approximation method selects the k spatially closest points to x_*, where k is some predefined number. For our work, we found this approach useful when dealing with the large, simulated data sets. It allowed us to evaluate our training methods on dense sets of simulation data in reasonable amounts of time. We did not apply this method in modeling the data from our sensor network as the training set was relatively small.

E. Refining the Model

Equation 3 can be used as the basis for refining the model; knowledge about the uncertainty of the initial predictions can be used to guide modifications to the placement of the sensor motes. By optimizing sensor placements, we aim to improve the model quality.

A common and intuitive technique is to place sensors at the locations with the highest variance ([16]). This is done in an iterative process. Beginning with no sensors placed, at each iteration the next sensor is placed at the location with the highest uncertainty, given the previous sensor placements. The process continues until all available sensors are placed. Unfortunately, this technique tends to place sensors far away from each other, resulting in placements along the edges of the space. This was noted by [17] and we also observed this in our own experiments. The result is poor prediction in the center of the space due to lack of sensor coverage.

A better placement technique is proposed by [18]. Their algorithm seeks to approximately maximize *mutual information*—rather than placing sensors at the most uncertain locations, the mutual information algorithm places sensors at the locations that best reduce uncertainty over the entire space. The algorithm is again iterative, greedily placing each sensor at the location that maximizes mutual information, given the previous sensor placements. As we are concerned with multiple distributions—for both temperature and humidity—we extended the algorithm to select locations that maximize mutual information for both distributions.

In this work we use the mutual information criterion to optimize sensor placements in the model refining step, as it produces the best results. Details of the algorithm are given in [18].

IV. EXPERIMENTS

We used simulated data for initial exploration into the model selection process. This allowed for easier automation

of the process. Figure 2 shows an example simulated temperature distribution. Simulated data was used to evaluate the effectiveness of different search techniques for learning the hyperparameters and to evaluate the suitability of the various kernel functions.

Based on initial experiments, we opted for a simple gradient descent search (Quasi-Newton with BFGS Hessian update) to learn the hyperparameters. We generated a set of feasible initial points and performed the optimization starting from each, retaining the best solution reached from the set of points. Knowledge of our particular environment suggested upper and lower bounds on the various hyperparameters. Initial start points were generated randomly across this likely space, as [19] argues that random search outperforms a grid search. To ensure even coverage of the space, we again used the Halton sequence for generating well distributed points.

Using this search method and equation 1, we computed the marginal likelihood for the covariance functions listed in Table I. Averaging the likelihood over numerous trials on simulated data sets, we obtained the results in Table II. The results give reason to expect the Matérn kernel to perform best in the real-world case.

TABLE II
KERNEL LIKELIHOOD

Kernel	Temperature	Humidity
Squared Exponential	-49.75	-53.33
Rational Quadratic	-63.24	-68.80
Matérn	-42.02	-49.45

Actual data collection was done using twenty-seven wireless sensor motes. Each mote was equipped with highly accurate temperature and relative humidity sensors (reported in degrees Fahrenheit and percent, respectively). Readings were taken continuously and logged for an extended period of time. For our experiments, twenty of the sensors were used for training points and distributed across the lab space as previously explained. The remaining seven were withheld from use in creating the models to be used as test points and were randomly distributed across the lab.

Using data collected from the initial sensor deployment, we computed the temperature and humidity distributions across the lab space. Figures 3 and 4 show the inferred temperature and humidity, respectively, across the lab. For ease of viewing, we plot just a horizontal slice of the distribution.

As a measure of the model performance, we calculated the mean squared error (MSE) of the model predictions at our seven test points. Table III shows the results for the temperature and humidity models based on the various kernels.

As expected, the Matérn kernel outperformed the others. Using these initial models, we adjusted the sensor placements as previously described to improve the model. Table IV shows the performance of the Matérn based model after being refined. Sensor placement according to the mutual information algorithm resulted in significant improvement.

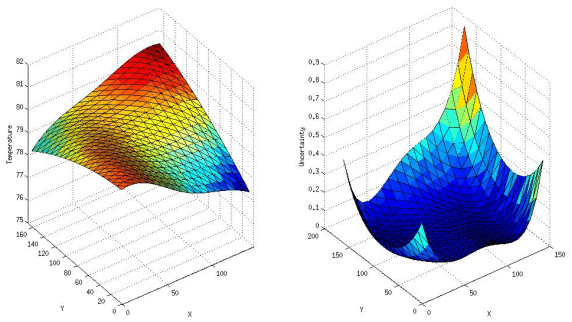

Fig. 3. Temperature model of actual lab space, mean (left) and variance (right). Values displayed are on the horizontal plane, midway between the ceiling and floor.

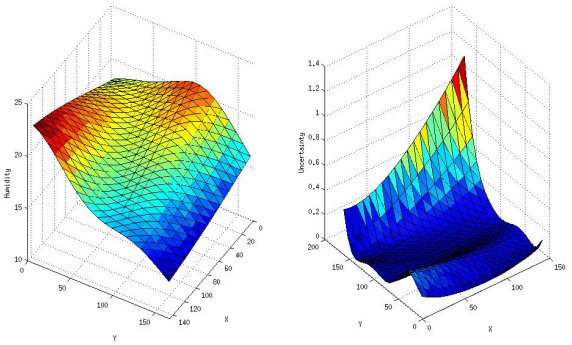

Fig. 4. Humidity model of actual lab space, mean (left) and variance (right). Values displayed are on the horizontal plane, midway between the ceiling and floor.

Besides using Gaussian processes to interpolate spatially distributed sensor readings, we also investigated the use of Gaussian processes in predicting environmental conditions over time. Using data logged by the sensors over a set time period (for this experiment, we collected data over a 400 second interval), we constructed a training set consisting of readings from the twenty training sensors at various times. Thus, the training set consisted of 4-dimensional inputs. Test data was collected similarly. Extending the techniques discussed previously to incorporate time as well as position, we created a Gaussian process model to predict environmental

TABLE III
INITIAL MODEL PERFORMANCE

Kernel	Temperature MSE	Humidity MSE
Squared Exponential	0.84 sq deg	2.28 sq pct
Rational Quadratic	1.07 sq deg	2.28 sq pct
Matérn	0.79 sq deg	1.36 sq pct

TABLE IV
REFINED MODEL PERFORMANCE

Kernel	Temperature MSE	Humidity MSE
Matérn	.42 sq deg	.67 sq pct

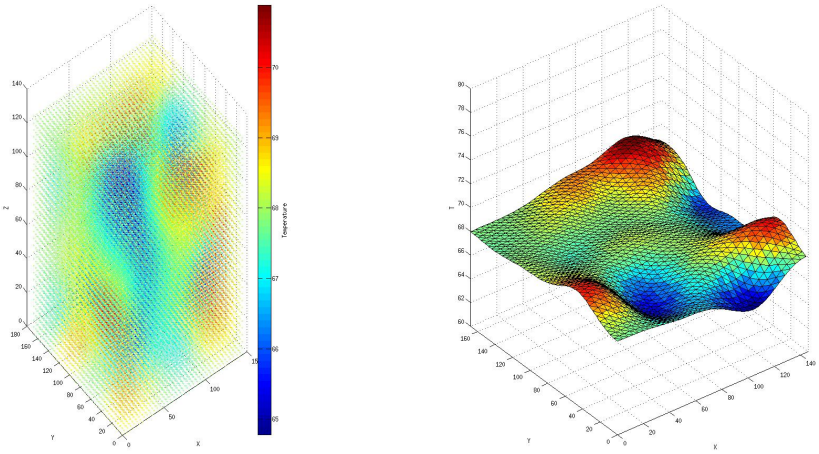

Fig. 2. Point cloud displaying simulated temperatures (left). Temperature on horizontal plane, midway between the ceiling and floor (right).

conditions across the space at any time. Figure 5 shows the test sensor readings at a single location along with the predicted temperature at that location, obtained from our Gaussian process model. Surrounding the predicted value, we show uncertainty bounds on the prediction computed using equation 3. For interpolation, the prediction performs reasonably well. Note though that beyond the 400 second mark—where no training data was collected—the uncertainty on the predictions grows rapidly. This characterization of uncertainty over time could be particularly applicable to dynamic monitoring systems in deciding when mobile robots ought to be deployed to gather additional readings.

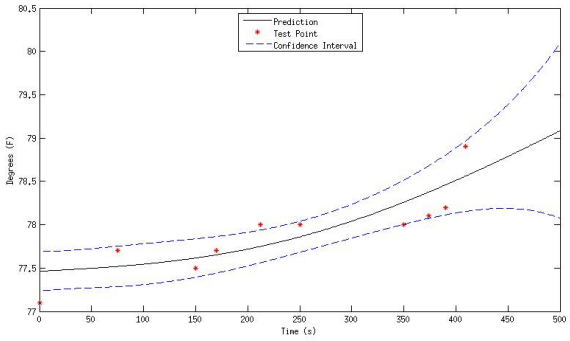

Fig. 5. Temperature prediction at single test point over time.

V. CONCLUSION

This work explored a novel approach to combining multisensory input to monitor multiple spatially distributed environmental variables. Specifically, it explored a Gaussian process based approach to interpolating discrete sensor data and showed that the approach produces meaningful and useful results, even on sparse data sets. Various covariance functions were evaluated and the Matérn kernel was shown to be a preferable alternative to the more common squared exponential and rational quadratic kernels. Various aspects of the model selection process were studied on simulated data sets. Deployment of a sensor network across an indoor lab space provided real-world data sets. Experiments performed on these data sets verified the utility of the Gaussian process approach. Uncertainty about predictions was used in conjunction with existing algorithms to improve sensor placement, resulting in more accurate environmental models. Sensor placement optimization was extended to consider different variables—each with its own model—monitored by the same sensor. Prediction of variables over time was considered. The final result and contribution of this paper is an approach to monitoring environmental conditions in a 3-dimensional space, applicable to statically placed sensor networks as well as mobile robot systems.

REFERENCES

[1] J. Valverde, V. Rosello, G. Mujica, J. Portilla, A. Uriarte, and T. Riesgo, "Wireless Sensor Network for Environmental Monitoring: Application in a Coffee Factory," *International Journal of Distributed Sensor Networks*, vol. 2012, 2012.

[2] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," in *Proceedings Wireless Sensor Networks and Applications*, Atlanta, GA, September 2002, pp. 88–97.

[3] K. Black, *Business Statistics*. New York: John Wiley & Sons, 2011.

[4] J. Lenchner, C. Isci, J. Kephart, C. Mansley, J. Connell, and S. McIntosh, "Towards Data Center Self-Diagnosis Using a Mobile Robot," in *Proceedings of the 8th ACM International Conference on Autonomic Computing*, Karlsruhe, Germany, 2011, pp. 81–90.

[5] C. Rasmussen, *Gaussian Processes for Machine Learning*. Camridge, MA: MIT Press, 2006.

[6] J. Halton, "Algorithm 247: Radical-inverse Quasi-random Point Sequence," *Communications of the ACM*, vol. 7, no. 12, pp. 70–71, 1964.

[7] M. Stein, *Interpolation of Spatial Data: Some Theory for Kriging*. Berlin, Germany: Springer Verlag, 1999.

[8] C.-S. Foo, B. Chuong, and A. Ng, "Efficient Multiple Hyperparameter Learning for Log-linear Models," in *NIPS 2007*, 2007, pp. 377–384.

[9] Y. Bengio, "Gradient-based Optimization of Hyper-parameters," *Neural Computation*, vol. 12, no. 8, pp. 1889–1900, 2000.

[10] M. Blum and M. Riedmiller, "Optimization of Gaussian Process Hyperparameters using Rprop," in *Proceedings of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, Bruges, BE, April 2013.

[11] A. Corana, M. Marchesi, C. Martini, and S. Ridella, "Minimizing Multimodal Functions of Continuous Variables with the Simulated Annealing Algorithm," *ACM Transactions on Mathematical Software*, vol. 13, no. 3, pp. 262–280, 1987.

[12] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Gaussian Process Modeling of Large-scale Terrain," *The Journal of Field Robotics*, vol. 26, no. 10, pp. 812–840, 2009.

[13] S. Vasudevan, F. Ramos, E. Nettleton, H. Durrant-Wyhte, and A. Blair, "Gaussian Process Modeling of Large-Scale Terrain," in *The Proceedings of the International Conference on Robotics and Automation*. Kobe, Japan: IEEE, 2009, pp. 1047–1053.

[14] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Wyhte, "Evaluation of Gaussian Processes for Large-Scale Terrain Modeling," in *The Proceedings of the Australasian Conference on Robotics and Automation*, Sydney, Australia, 2009.

[15] S. Vasudevan, F. Ramos, E. Nettleton, H. Durrant-Wyhte, and A. Blair, "Large-scale Terrain Modeling from Mulitple Sensors with Dependent Gaussian Processes," in *The Proceedings of the International Conference on Intelligent Robots and Systems*. Taipei, Taiwan: IEEE, 2010, pp. 1215–1221.

[16] N. Cressie and N. Cassie, *Statistics for Spatial Data*. Wiley New York, 1993, vol. 900.

[17] N. Ramakrishnan, C. Bailey-Kellogg, S. Tadepalli, and V. Pandey, "Gaussian Processes for Active Data Mining of Spatial Aggregates," in *SDM*. SIAM, 2005, pp. 427–438.

[18] C. Guestrin, A. Krause, and A. Singh, "Near-Optimal Sensor Placements in Gaussian Processes," in *Proceedings of the 22nd International Conference on Machine learning*. ACM, 2005, pp. 265–272.

[19] J. Bergstra and Y. Bengio, "Random Search for Hyper-parameter Optimization," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 281–305, 2012.