Spatial Gaussian Process Regression With Mobile Sensor Networks

Dongbing Gu, *Senior Member, IEEE*, and Huosheng Hu, *Senior Member, IEEE*

***Abstract*— This paper presents a method of using Gaussian process regression to model spatial functions for mobile wireless sensor networks. A distributed Gaussian process regression (DGPR) approach is developed by using a sparse Gaussian process regression method and a compactly supported covariance function. The resultant formulation of the DGPR approach only requires neighbor-to-neighbor communication, which enables each sensor node within a network to produce the regression result independently. The collective motion control is implemented by using a locational optimization algorithm, which utilizes the information entropy from the DGPR result. The collective mobility of sensor networks plus the online learning capability of the DGPR approach also enables the mobile sensor network to adapt to spatiotemporal functions. Simulation results are provided to show the performance of the proposed approach in modeling stationary spatial functions and spatiotemporal functions.**

***Index Terms*— Coverage control, Gaussian process regression (GPR), mobile sensor networks, spatiotemporal modeling.**

I. Introduction

ENVIRONMENTAL surveillance in many areas, such as meteorology, climatology, ecology, oceanography, etc., requires the capability of modeling spatial functions, or even spatiotemporal functions. The distributed nature and the collective mobility of mobile sensor networks offer a feasible solution to environmental surveillance applications by means of collectively sampling data and cooperatively modeling spatial functions. Several research projects have targeted to this research area, such as monitoring forest fires using unmanned aerial vehicles (UAVs) in [1], monitoring air quality using UAVs in [2], monitoring ocean ecology conditions using unmanned underwater vehicles in [3].

A mobile sensor network is able to make sensory observations of environmental physical phenomena with on-board sensors, exchange information via on-board wireless communication devices, and explore over the area of interest with the mobility. As a result, it is able to produce a predictive model for the sampled physical phenomenon and allocate itself in a pattern which can generate a more accurate predictive model. The strategies of cooperative modeling and coordinated motion are the key techniques to be developed.

Gaussian process regression (GPR), also known as the Kriging filter, is a popular regression technique for data assimilation. In recent years, there has been a growing interest in Gaussian processes for regression and classification tasks [4]–[6]. A GPR is specified by a predictive mean function, a predictive covariance function, and a set of hyperparameters which can be determined from sampled data [7]–[9]. The main advantage of the GPR over other regression techniques is its ability to predict not only the mean function of the Gaussian process (GP), but also the covariance function of the GP. The predicted uncertainty is valuable information for further decision-making in environmental surveillance applications. In [10], GPR was applied for monitoring the ecological condition of a river for individual sensors. The sensor placement was determined by maximizing a mutual information gain. In [11], the computation complexity of the GPR was reduced by a Bayesian Monte Carlo approach, and an information entropy was used to allocate individual sensors. In [12], a mixture of GPs was applied for building a gas distribution system, with the aim of reducing the computation complexity. In [13], a Kalman filter was built on the top of a GP model to characterize spatiotemporal functions. A path planning problem was solved by optimizing a mutual information gain. In [14], a Kriged Kalman filter was developed to reconstruct spatiotemporal functions and a centroidal Voronoi tessellation (CVT) algorithm was employed to allocate mobile sensor nodes. The Kriged Kalman filter and swarm control were used in [15] to reconstruct spatiotemporal functions. In [16], several nonseparable spatiotemporal covariance functions were proposed for modeling spatiotemporal functions.

Environmental spatial functions have been modeled in mobile sensor networks by using radial basis function (RBF) networks in [17] where a CVT coverage control was used to allocate mobile sensor nodes, and in [18] where a flocking control was used to move sensor nodes. The environmental spatial function was approximated by using an inverse distance weighting interpolation method and updated by using a Kalman filter in [19]. In the function space view of GPR [7], RBF is regarded as a truncated GP model where a limited number of basis functions are used.

A centralized GPR can be implemented in a sensor network by sending data from all sensor nodes to a centralized unit where the computation of the predictive mean function and the covariance function is conducted. For making effective motion

Manuscript received January 17, 2011; revised May 14, 2012; accepted May 15, 2012. Date of publication June 15, 2012; date of current version July 16, 2012. This work was supported in part by the EU FP7 Program, ICT-231646, and SHOAL.

The authors are with the School of Computer Science and Electronic Engineering, University of Essex, Essex CO4 3SQ, U.K. (e-mail: dgu@essex.ac.uk; hhu@essex.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2012.2200694

decisions, the predictive results need to be sent back to each sensor node. Thus this version of GPR could cause a severe problem for wireless communication. It is also not robust to the failure of individual nodes. Further, it does not scale well with the size of networks, as GPR is a nonparametric regression technique and its computational complexity grows in the order of $\mathcal{O}(N^3)$ with the size N of sampled data. In the GPR research community, various sparse Gaussian process regression (SGPR) techniques have been proposed to alleviate the computation complexity problem [9], [20]. They are based on a small set of active points which are projected from the sampled data according to variant projection strategies under distribution approximation assumptions.

In this paper, an SGPR strategy, called the projected process approximation in [21] and [22], is used to develop a distributed Gaussian process regression (DGPR). In this proposed DGPR approach, the active point set of the SGPR corresponds to the neighbor set definition of sensor networks. It also uses a compactly supported covariance function to decouple the computation in a local node from the contribution of remote nodes. A compactly supported covariance function, called the Wu's polynomial positive-definite function [23], is adopted in this paper, as its analytic derivative form is available for the hyperparameter learning. The hyperparameters of covariance function are learned by maximizing the marginal likelihood of the sampled data. The proposed DGPR approach can make this learning algorithm a distributed algorithm so that a mobile sensor network can update the predictive mean function and covariance function online. This feature extends the proposed DGPR approach from modeling spatial functions to modeling spatiotemporal functions. Because of its distributed nature, it is robust against the failure of individual nodes and scales well with the size of networks. The collective motion strategy is based on the predictive result of the GP. The CVT algorithm is adopted for motion coordination. However, the utility function of CVT algorithm is not defined as a mean function as others. It is defined as an information entropy. Use of the predictive mean utility function results in the sensor nodes moving to the locations where high mean values are predicted. This could lead to the sensor nodes clustering on some locations where the predictions are very certain while some locations where the predictions are poor are unobserved. With the information entropy utility function, the sensor nodes potentially move to the locations where high uncertainties are predicted and then make observations there to reduce the uncertainties.

In summary, this paper contributes to the development of a novel DGPR algorithm which is built on an SGPR strategy combined with a compactly supported covariance function. Given the distributed implementation of GPR, information entropy is available for the decision-making in collective motion control process for each sensor node by using information entropy as the utility function in the CVT algorithm. In the following, Section II revisits some basics of GPR and presents the SGPR strategy within the framework of a mobile sensor network. The compactly supported covariance function and the proposed DGPR are detailed in Section III. The information entropy-based coverage control is introduced in Section IV. Section V provides the simulation results. Our conclusion and future work are given in Section VI.

II. GPR

A mobile sensor network with N sensors is to be deployed in a 2-D area Q to model a scalar environmental spatial function in that area. The sensor node i is located at a 2-D position $\mathbf{x}_{i,t}$ and it is assumed that the position $\mathbf{x}_{i,t}$ can be found by itself with self-localization techniques at time step t. Each sensor node i can make a point observation $z_{i,t}$ of an environmental spatial function $f(\mathbf{x}_{i,t})$ at time step t. The sensory observation distribution is assumed to be Gaussian

$$z_{i,t} = f(\mathbf{x}_{i,t}) + \varepsilon_{i,t}$$

where $\varepsilon_{i,t}$ is a Gaussian noise with zero mean and covariance $\sigma_{i,t}^2$, noted as $\varepsilon_{i,t} = \mathcal{N}(0, \sigma_{i,t}^2)$.

In this section, it is assumed that each sensor node can collect the location information $\mathbf{x}_{j,t}$ and the corresponding observation $z_{j,t}$ from all the other sensor nodes via wireless communication. The distributed implementation will be detailed in the next section.

A. GPR

In a sensor node, the Gaussian inference is conducted at each time step based on the information available at that moment. The given information in a sensor node includes the input vectors $X_{N,t} = [\mathbf{x}_{1,t}, \ldots, \mathbf{x}_{N,t}]^T$ and the corresponding observations $\mathbf{z}_{N,t} = [z_{1,t}, \ldots, z_{N,t}]^T$.

In a GP model, the prior distribution of latent variable $f_{i,t} = f(\mathbf{x}_{i,t})$ is modeled as a Gaussian. Its mean value is assumed to be zero because offsets and simple trends can be subtracted out first. The prior knowledge about multiple latent variables is modeled by a covariance matrix $K_{NN} = [k(\mathbf{x}_{i,t}, \mathbf{x}_{j,t})]$, where $k(\mathbf{x}_{i,t}, \mathbf{x}_{j,t})$ is a covariance function. With a positive-definite covariance matrix K_{NN}, the GP prior distribution of full latent variables $\mathbf{f}_{N,t} = [f_{1,t}, \ldots, f_{N,t}]^T$ is represented as

$$p(\mathbf{f}_{N,t}) = \mathcal{N}(\mathbf{0}, K_{NN}).$$

The likelihood distribution of the observation vector $\mathbf{z}_{N,t}$ is represented as

$$p(\mathbf{z}_{N,t}|\mathbf{f}_{N,t}) = \mathcal{N}(\mathbf{f}_{N,t}, R_{N,t}) \tag{1}$$

where $R_{N,t} = \text{diag}(\sigma_{1,t}^2, \ldots, \sigma_{N,t}^2)$.

GPR can infer $f_{*,t} = f(\mathbf{x}_{*,t})$ for a test point $\mathbf{x}_{*,t} \in Q$ using $p(f_{*,t}|\mathbf{z}_{N,t})$ given a training data set $(X_{N,t}, \mathbf{z}_{N,t})$. The latent predictive distribution of a given test point is obtained by solving the maximum *a posteriori* problem and is given by

$$p(f_{*,t}|\mathbf{z}_{N,t}) = \mathcal{N}(\mu_{*,t}, \Sigma_{*,t}) \tag{2}$$

where the predictive mean function and the predictive covariance function are

$$\begin{aligned} \mu_{*,t} &= K_{*N}(K_{NN} + R_{N,t})^{-1}\mathbf{z}_{N,t} \\ \Sigma_{*,t} &= K_{**} - K_{*N}(K_{NN} + R_{N,t})^{-1}K_{N*} \end{aligned} \tag{3}$$

where $K_{*N} = [k(\mathbf{x}_{*,t}, \mathbf{x}_{1,t}), \ldots, k(\mathbf{x}_{*,t}, \mathbf{x}_{N,t})]$ and $K_{**} = k(\mathbf{x}_{*,t}, \mathbf{x}_{*,t})$. $K_{N*} = K_{*N}^T$ for symmetrical covariance functions.

B. SGPR

For agent i, the full dataset is separated into a neighbor set N_i which includes i itself, and a nonneighbor set $\bar{N}_i$. The neighbor set N_i is viewed as the active set and the nonneighbor set $\bar{N}_i$ is viewed as the remaining set of the SGPR with projected process approximation [22].

In a GPR, the prior distribution of full latent variables can be rewritten into a form with the separate sets as follows:

$$p(\mathbf{f}_{N_i,t}, \mathbf{f}_{\bar{N}_i,t}) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K_{N_iN_i} & K_{N_i\bar{N}_i} \\ K_{\bar{N}_iN_i} & K_{\bar{N}_i\bar{N}_i} \end{bmatrix}\right). \quad (4)$$

From above, the conditional distribution can be found as

$$p(\mathbf{f}_{\bar{N}_i,t}|\mathbf{f}_{N_i,t}) = \mathcal{N}\Big(K_{\bar{N}_iN_i}K_{N_iN_i}^{-1}\mathbf{f}_{N_i,t} \times K_{\bar{N}_i\bar{N}_i} - K_{\bar{N}_iN_i}K_{N_iN_i}^{-1}K_{N_i\bar{N}_i}\Big). \quad (5)$$

However in an SGPR, the prior distribution of the full latent variables in (4) is not available. Only the prior distribution of latent variables in the active set N_i is assumed to be Gaussian

$$p(\mathbf{f}_{N_i,t}) = \mathcal{N}(\mathbf{0}, K_{N_i,N_i}). \quad (6)$$

Without the prior distribution of full latent variables in (4), the conditional distribution in (5) cannot be directly obtained. There are several approximation approaches available. Among them, the projected process approximation approach is an effective way (see details in [9]).

The SGPR with projected process approximation is recovered in a unifying framework of SGPR in [20], where the SGPR is deduced by using two GPRs. In the first GPR, the conditional distribution in (5) is approximated by a delta function which keeps the mean unchanged

$$p(\mathbf{f}_{\bar{N}_i,t}|\mathbf{f}_{N_i,t}) = \mathcal{N}(K_{\bar{N}_iN_i}K_{N_iN_i}^{-1}\mathbf{f}_{N_i,t}, \mathbf{0}). \quad (7)$$

This approximation, combined with the fact that

$$\begin{aligned} p(\mathbf{f}_{N_i,t}|\mathbf{f}_{N_i,t}) &= \mathcal{N}(\mathbf{f}_{N_i,t}, \mathbf{0}) \\ &= \mathcal{N}(K_{N_iN_i}K_{N_iN_i}^{-1}\mathbf{f}_{N_i,t}, \mathbf{0}) \end{aligned} \quad (8)$$

contributes a conditional distribution of full latent variables

$$p(\mathbf{f}_{N,t}|\mathbf{f}_{N_i,t}) = \mathcal{N}(K_{NN_i}K_{N_iN_i}^{-1}\mathbf{f}_{N_i,t}, \mathbf{0}) \quad (9)$$

where $K_{N_iN} = [K_{N_iN_i} \;\; K_{N_i\bar{N}_i}]$ and $K_{NN_i} = K_{N_iN}^T$.

Given the prior distribution of latent variables in neighbor set (6), the joint distribution of latent variable with a test point $f_{*i,t}$ is known as follows:

$$p(\mathbf{f}_{N_i,t}, f_{*i,t}) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K_{N_iN_i} & K_{N_i*} \\ K_{*N_i} & K_{**} \end{bmatrix}\right). \quad (10)$$

From above joint distribution in (10), the conditional distribution of a testing point in the first GPR is obtained as

$$p(f_{*i,t}|\mathbf{f}_{N_i,t}) = \mathcal{N}\Big(K_{*N_i}K_{N_iN_i}^{-1}\mathbf{f}_{N_i,t}, K_{**} - K_{*N_i}K_{N_iN_i}^{-1}K_{N_i*}\Big). \quad (11)$$

Given the latent variables of a neighbor set, it is assumed that $\mathbf{f}_{N,t}$ and $f_{*i,t}$ are independent. Then the effective prior distribution is given by marginalizing out the neighbor latent variables

$$\begin{aligned} p(\mathbf{f}_{N,t}, f_{*i,t}) &= \int p(\mathbf{f}_{N,t}|\mathbf{f}_{N_i,t})p(f_{*i,t}|\mathbf{f}_{N_i,t})p(\mathbf{f}_{N_i,t})d\mathbf{f}_{N_i,t} \\ &= \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} Q_{NN} & Q_{N*} \\ Q_{*N} & K_{**} \end{bmatrix}\right) \end{aligned} \quad (12)$$

where

$$\begin{aligned} Q_{NN} &= K_{NN_i}K_{N_iN_i}^{-1}K_{N_iN} \\ Q_{N*} &= K_{NN_i}K_{N_iN_i}^{-1}K_{N_i*} \\ Q_{*N} &= K_{*N_i}K_{N_iN_i}^{-1}K_{N_iN}. \end{aligned} \quad (13)$$

In the second GPR, by combining the effective prior distribution in (12) with the likelihood distribution in (1), the joint distribution between observations and a test point latent variable is given as follows:

$$p(\mathbf{z}_{N,t}, f_{*i,t}) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} Q_{NN} + R_{N,t} & Q_{N*} \\ Q_{*N} & K_{**} \end{bmatrix}\right). \quad (14)$$

From the joint distribution in (14), the posterior conditional distribution is obtained as follows:

$$p(f_{*i,t}|\mathbf{z}_{N,t}) = \mathcal{N}(\mu_{*i,t}, \Sigma_{*i,t}) \quad (15)$$

where

$$\begin{aligned} \mu_{*i,} &= Q_{*N}(Q_{NN} + R_{N,t})^{-1}\mathbf{z}_{N,t} \\ \Sigma_{*i,t} &= K_{**} - Q_{*N}(Q_{NN} + R_{N,t})^{-1}Q_{N*}. \end{aligned} \quad (16)$$

Remark 1: Although the active set or the neighbor set is used in SGPR and the computation complexity can be reduced by using the matrix inversion lemma for $(Q_{NN} + R_{N,t})^{-1}$, the predictive mean function and the predictive covariance function in a node i still require all the other nodes to send their data to it. To develop a DGPR, a further requirement, for instance a compactly supported covariance function, is necessary.

III. DGPR

The prior knowledge about $k(\mathbf{x}_{i,t}, \mathbf{x}_{j,t})$ is the crucial ingredient in a GPR. It determines the properties of underlying process. We start this section with a general covariance function. Then a compactly supported covariance function is introduced. In the end, the core of this paper, DGPR, is formulated.

A. Matern Family of Covariance Functions

The Matern family of covariance functions is a class of functions that are flexible and commonly used in data assimilation. It is defined as follows:

$$\begin{aligned} k(\mathbf{x}_{i,t}, \mathbf{x}_{j,t}) &= k(r_t) \\ &= \frac{a_t}{2^{\nu-1}\Gamma(\nu)}\left(\frac{\sqrt{2\nu}r_t}{l_t}\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}r_t}{l_t}\right) \end{aligned} \quad (17)$$

where Γ is the Gamma function and K_{ν} is the modified Bessel function of the second-order ν [24, pp. 374–379]. $r_t = \|\mathbf{x}_{i,t} - \mathbf{x}_{j,t}\|$. a_t is the amplitude. l_t is the effective range. A smaller

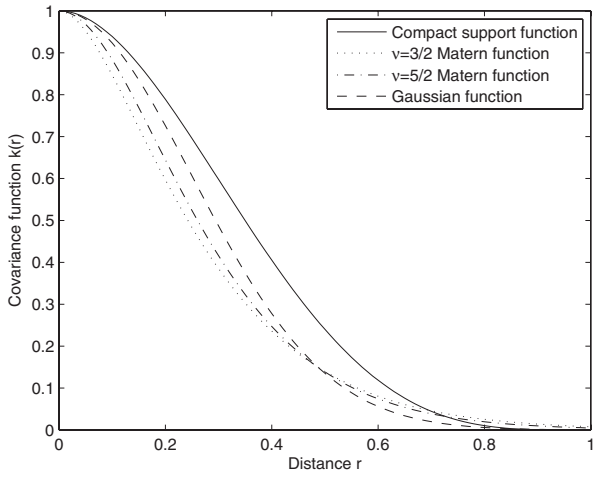


Fig. 1. Covariance functions.

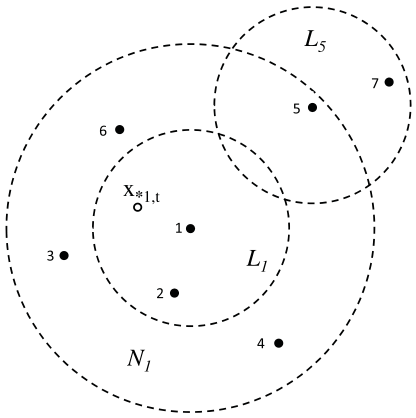


Fig. 2. Neighbor sets.

effective range implies the sample function of GP model varies more rapidly and a larger effective range implies the sample function of GP model varies more slowly.

ν is a smoothness parameter and its integer part determines the number of times the underlying spatial process is mean-square differentiable [9]. When $\nu = 1/2$, the function defined in (17) is an exponential function. When $\nu \to \infty$, the function defined in (17) converges to the squared exponential covariance function or the Gaussian covariance function

$$k(\mathbf{x}_{i,t}, \mathbf{x}_{j,t}) = a_t e^{-\frac{r_t^2}{2l_t^2}}. \tag{18}$$

The Gaussian covariance function is infinite-differentiable, which means that the GP with this covariance function has mean square derivatives of all orders, and is thus very smooth. However, such a strong smoothness assumption is unrealistic for modeling some physical processes [25].

When ν is of the form $m + 1/2$ with m a nonnegative integer, K_ν takes an explicit form. In this case, the covariance function is a product of an exponential and a polynomial of order m. When $\nu < 1$, the covariance function is continuous, but nondifferential. When $\nu \geq 7/2$, the covariance functions are all very similar to the Gaussian covariance function. Thus there are two values of ν that are useful, i.e., $\nu = 3/2$ or $\nu = 5/2$. For $\nu = 3/2$

$$k(\mathbf{x}_{i,t}, \mathbf{x}_{j,t}) = a_t e^{-\frac{\sqrt{3}r_t}{l_t}} \left(\frac{\sqrt{3}r_t}{l_t} + 1 \right). \tag{19}$$

The infinite support of these covariance functions leads to computational complexity because of the demand of computing the inverse of covariance matrix in the order of $\mathcal{O}(N^3)$.

B. Compactly Supported Covariance Functions

The sparse covariance matrix is favorable to reducing the computational complexity of the GPR. Covariance tapering is one way to taper a covariance matrix with an appropriate compactly supported covariance function, which can reduce the computational burden significantly and still lead to asymptotically optimal mean-squared error [26]. Another option is to directly use a compactly supported covariance function. Our proposed DGPR is built on the use of a compactly supported covariance function so that only local communication is required. There are several positive-definite covariance functions with compact support reported in [27]. In this paper, we use Wu's polynomial function [23], [27] with the following expression:

$$\begin{aligned} k(\mathbf{x}_{i,t}, \mathbf{x}_{j,t}) &= k(r_t) \\ &= a_t \left(1 - \frac{r_t}{l_t}\right)_+^4 \left(1 + \frac{4r_t}{l_t} + \frac{3r_t^2}{l_t^2} + \frac{3r_t^3}{4l_t^3}\right) \end{aligned} \tag{20}$$

where

$$\left(1 - \frac{r_t}{l_t}\right)_+ = \begin{cases} \left(1 - \frac{r_t}{l_t}\right), & \text{if } r_t < l_t \\ 0, & \text{otherwise.} \end{cases}$$

This polynomial function is with C^2 smoothness. The smoothness is a guarantee for good regression results and the compact support is a requirement for the distributed DGR. Other high-order Wu's polynomial functions look similar to this one, but take more time to compute. The comparisons with infinitely supported covariance functions in (18) and (19) are made in Fig. 1, where $a_t = 1$ and $l_t = 0.3$ are used for the infinitely supported covariance functions and $a_t = 1$ and $l_t = 1.0$ are used for the compactly supported covariance function in (20). It can be seen that the compactly supported covariance function preserves some shape of the infinitely supported covariance functions.

Remark 2: In general, a compactly supported covariance function means that the distant observation will have zero effect on the prediction result. In some cases where a covariance with long-range correlation is effective, it should still be useful qualitatively. There is an unproved argument in geostatistics that the prediction result can be nearly independent of distant observation conditional on the neighbors even if it may be highly correlated with distant observation [26].

C. DGPR

Let D_i denote the communication range for node i. The neighbor set N_i used in the SGPR is termed as the

communication range neighbor set N_i and is defined as follows:

$$N_i = \{j \in N \mid \|\mathbf{x}_{i,t} - \mathbf{x}_{j,t}\| \leq D_i\} \cup i.$$

For each node i, the compactly supported covariance function is defined within the local region and therefore has local hyperparameters $l_{i,t}, a_{i,t}$. An effective range neighbor set L_i can be defined for node i as follows:

$$L_i = \{j \in N \mid \|\mathbf{x}_{i,t} - \mathbf{x}_{j,t}\| \leq l_{i,t}\} \cup i.$$

A prediction neighbor set M_i of node i will be useful for DGPR presentation and is defined as follows:

$$M_i = \bigcup_{j=1}^{|N_i|} L_j$$

where $|N_i|$ is the cardinality of set N_i. Let $\mathbf{x}_{*i,t}$ be a testing point of node i. It is within the area defined by $\{\mathbf{x} \in Q \mid \|\mathbf{x}_{i,t} - \mathbf{x}\| \leq l_{i,t}\}$. It should be noted that $D_i \geq 2l_{i,t}$ is required so that $k(\mathbf{x}_{*i,t}, \mathbf{x}_{j,t}) = 0$ for $j \notin N_i$.

The neighbor sets of sensor nodes (black dots) defined above are depicted in Fig. 2. L_1 is the effective range neighbor set and N_1 is the communication range neighbor set of node 1. $L_1 = \{1, 2\}$. $L_5 = \{5, 7\}$. $N_1 = \{1, 2, 3, 4, 5, 6\}$. $M_1 = \{1, 2, 3, 4, 5, 6, 7\}$.

Now we are in a position to deduce the DGPR from the SGPR in (16). Given the neighbor set definitions and the compactly supported covariance function in (20), the following results are obtained.

1) The computation of $K_{N_iN_i}$ for node i only requires that its communication range neighbor j ($j \in N_i$) sends the position $\mathbf{x}_{j,t}$ and observation $z_{j,t}$ to it.
2) The computation of K_{*N_i} for node i only requires that its communication range neighbor j ($j \in N_i$) sends the position $\mathbf{x}_{j,t}$ and observation $z_{j,t}$ to it. This is possible due to the requirement of $D_i \geq 2l_{i,t}$.
3) The computation of K_{N_iN} for node i requires that its prediction range neighbor j ($j \in M_i$) sends the position $\mathbf{x}_{j,t}$ and observation $z_{j,t}$ to it. This becomes clear by rearranging the matrix rows and columns so that its first M_i columns are from the set M_i and all the remaining columns have $k(x_{i,t}, x_{j,t}) = 0$ for $j \notin M_i$, i.e., $K_{N_iN} = [K_{N_iM_i} \quad \mathbf{0}]$.

Next the following matrices are defined:

$$\begin{aligned} Q_{M_iM_i} &= K_{M_iN_i} K_{N_iN_i}^{-1} K_{N_iM_i} \\ Q_{M_i*} &= K_{M_iN_i} K_{N_iN_i}^{-1} K_{N_i*} \\ Q_{*M_i} &= K_{*N_i} K_{N_iN_i}^{-1} K_{N_iM_i}. \end{aligned} \tag{21}$$

Using the above matrices, the SGPR matrices defined in (13) are changed into

$$\begin{aligned} Q_{*N} &= K_{*N_i} K_{N_iN_i}^{-1} [K_{N_iM_i} \quad \mathbf{0}] = [Q_{*M_i} \quad \mathbf{0}] \\ Q_{N*} &= \begin{bmatrix} K_{M_iN_i} \\ \mathbf{0} \end{bmatrix} K_{N_iN_i}^{-1} K_{N_i*} = \begin{bmatrix} Q_{M_i*} \\ \mathbf{0} \end{bmatrix} \\ Q_{NN} &= \begin{bmatrix} K_{M_iN_i} \\ \mathbf{0} \end{bmatrix} K_{N_iN_i}^{-1} [K_{N_iM_i} \quad \mathbf{0}] \\ &= \begin{bmatrix} Q_{M_iM_i} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \end{aligned}$$

Finally, the predictive mean function and the predictive covariance function of DGPR are found as follows:

$$\begin{aligned} \mu_{*i,t} &= Q_{*N}(Q_{NN} + R_{N,t})^{-1} \mathbf{z}_{N,t} \\ &= [Q_{*M_i} \quad \mathbf{0}] \\ &\quad \left(\begin{bmatrix} Q_{M_iM_i} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} R_{M_i,t} & \mathbf{0} \\ \mathbf{0} & * \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{z}_{M_i,t} \\ * \end{bmatrix} \\ &= Q_{*M_i}(Q_{M_iM_i} + R_{M_i,t})^{-1} \mathbf{z}_{M_i,t} \\ \Sigma_{*i,t} &= K_{**} - Q_{*N}(Q_{NN} + R_{N,t})^{-1} Q_{N*} \\ &= K_{**} - [Q_{*M_i} \quad \mathbf{0}] \\ &\quad \left(\begin{bmatrix} Q_{M_iM_i} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} R_{M_i,t} & \mathbf{0} \\ \mathbf{0} & * \end{bmatrix} \right)^{-1} \begin{bmatrix} Q_{M_i*} \\ \mathbf{0} \end{bmatrix} \\ &= K_{**} - Q_{*M_i}(Q_{M_iM_i} + R_{M_i,t})^{-1} Q_{M_i*}. \end{aligned} \tag{22}$$

Remark 3: The computation of DGPR asks for information from the nodes in the prediction range neighbor set M_i. This can be achieved by communicating with the nodes in the communication range neighbor set N_i because any neighbor j in N_i has information from the nodes in its effective range neighbor set L_j.

D. Hyperparameter Learning

The hyperparameter set of GP in node i is defined as $\theta_{i,t} = [\sigma_{i,t}, a_{i,t}, l_{i,t}]^T$. Given a hyperparameter set $\theta_{i,t}$, the log marginal likelihood is

$$\begin{aligned} \mathcal{L}_{i,t} &= -\log p(\mathbf{z}_{M_i,t} | \theta_{i,t}) \\ &= \frac{1}{2} \mathbf{z}_{M_i,t}^T C_{i,t}^{-1} \mathbf{z}_{M_i,t} + \frac{1}{2} \log \det(C_{i,t}) + \frac{|M_i|}{2} \log(2\pi) \end{aligned}$$

where $C_{i,t} = Q_{M_iM_i} + R_{M_i,t}$. The partial derivative is

$$\begin{aligned} \frac{\partial \mathcal{L}_{i,t}}{\partial \theta_{i,t}} &= -\frac{1}{2} \mathbf{z}_{M_i,t}^T C_{i,t}^{-1} \frac{\partial C_{i,t}}{\partial \theta_{i,t}} C_{i,t}^{-1} \mathbf{z}_{M_i,t} + \frac{1}{2} tr\left(C_{i,t}^{-1} \frac{\partial C_{i,t}}{\partial \theta_{i,t}} \right) \\ &= -\frac{1}{2} tr\left(\left(\alpha_{i,t} \alpha_{i,t}^T - C_{i,t}^{-1} \right) \frac{\partial C_{i,t}}{\partial \theta_{i,t}} \right) \end{aligned}$$

where $\alpha_{i,t} = C_{i,t}^{-1} \mathbf{z}_{M_i,t}$.

Although $C_{i,t} = Q_{M_iM_i} + R_{M_i,t}$ is analytically available, its partial derivatives with respect to the parameters $a_{i,t}, l_{i,t}$ are complex and its computation is time consuming. Here, $C_{i,t}$ is approximated by using $C_{i,t} = K_{M_iM_i} + R_{M_i,t}$. This approximation means that a GPR with M_i data for node i is used to learn the hyperparameters via the maximum likelihood learning approach.

With $C_{i,t} = K_{M_iM_i} + R_{M_i,t}$, the following results are obtained:

$$\begin{aligned} \frac{\partial C_{i,t}}{\partial \sigma_{i,t}} &= \left[\frac{\partial k(r_t)}{\partial \sigma_{i,t}} \right] = \left[2\sigma_{i,t} \right] \\ \frac{\partial C_{i,t}}{\partial a_{i,t}} &= \left[\frac{\partial k(r_t)}{\partial a_{i,t}} \right] = \left[\frac{2k(r_t)}{a_{i,t}} \right] \\ \frac{\partial C_{i,t}}{\partial l_{i,t}} &= \left[\frac{\partial k(r_t)}{\partial l_{i,t}} \right]. \end{aligned}$$

For the Gaussian covariance function (18), it can be found that

$$\frac{\partial k(r_t)}{\partial l_{i,t}} = \frac{a_{i,t} r_t^2}{l_{i,t}^3} e^{-\frac{r_t^2}{2l_{i,t}^2}}.$$

For the Matern covariance function with $\nu = 3/2$ (19), it can be found that

$$\frac{\partial k(r_t)}{\partial l_{i,t}} = \frac{3a_{i,t}r_t^2}{l_{i,t}^3} e^{-\frac{\sqrt{3}r_t}{l_{i,t}}}.$$

For the compactly supported covariance function (20), it can be found that

$$\frac{\partial k(r_t)}{\partial l_{i,t}} = \frac{4r_t k(r_t)}{l_{i,t}(l_{i,t} - r_t)_+} - a_{i,t}\left(1 - \frac{r_t}{l_{i,t}}\right)_+^4 \left(\frac{4r_t}{l_{i,t}^2} + \frac{6r_t^2}{l_{i,t}^3} + \frac{9r_t^3}{4l_{i,t}^4}\right).$$

The constraint on $\theta_{i,t}$ should be imposed with $\theta_{\min} \le \theta_{i,t} \le \theta_{\max}$, especially for $l_{i,t}$ and $l_{\max} = D_i/2$.

The maximum likelihood learning algorithm is based on the gradient formulation given above. At time step t, the parameter vector $\theta_{i,t} = [\sigma_{i,t}, a_{i,t}, l_{i,t}]^T$ is the initial value for the maximum likelihood learning algorithm. The estimated result is denoted as $\theta_{i,t}^* = [\sigma_{i,t}^*, a_{i,t}^*, l_{i,t}^*]^T$. In order to smooth the transition from one GP model to another, $\theta_{i,t}^*$ is not directly used for regression. Instead, it is fed into a low-pass filter to filter out the high-frequency noise. The output $\theta_{i,t+1}$ of the low-pass filter is used for regression at time step t. The low-pass filter is expressed as follows:

$$\begin{aligned} \sigma_{i,t+1} &= \lambda_\sigma \sigma_{i,t} + (1 - \lambda_\sigma)\sigma_{i,t}^* \\ a_{i,t+1} &= \lambda_a a_{i,t} + (1 - \lambda_a)a_{i,t}^* \\ l_{i,t+1} &= \lambda_l l_{i,t} + (1 - \lambda_l)l_{i,t}^* \end{aligned} \tag{23}$$

where $\lambda_\sigma, \lambda_a,$ and λ_l are updating constants in the range of $(0, 1)$.

Remark 4: The hyperparameter $\theta_{i,t}$ is updated according to (23). It means that the covariance function $k(\mathbf{x}_{i,t}, \mathbf{x}_{j,t})$ is nonstationary in both spatial and temporal domains.

IV. Information Entropy-Based Coverage Control

A sensor node i moves from current position $\mathbf{x}_{i,t}$ to next position $\mathbf{x}_{i,t+1}$ according to the point motion model

$$\mathbf{x}_{i,t+1} = \mathbf{x}_{i,t} + \mathbf{u}_{i,t}$$

where $\mathbf{u}_{i,t}$ is the control signal to be found. In this section, the strategy of finding the control signal in order to gain a better regression result is sought.

A. Information Entropy

Mobile sensor node i makes observation $z_{i,t}$ at its location $\mathbf{x}_{i,t}$, and obtains observation $z_{j,t}$ and position $\mathbf{x}_{j,t}$ from all the neighbors $j \in M_i$ via wireless communication. It then updates the hyperparameter $\theta_{i,t}$. With the DGPR, the evaluations of $\mu_{*i,t}$ and $\Sigma_{*i,t}$ are available.

One motion control strategy is to move sensor nodes toward the locations with high mean values $\mu_{*i,t}$. The consequence is a concentration of sensor nodes on the area with high mean values. However, this result cannot reduce the uncertainty of the GP model in other areas. Another strategy is to move the sensor nodes toward the locations with high uncertainty $\Sigma_{*i,t}$. The consequence is a concentration of sensor nodes on the area with high covariance values. Then sensor nodes can make observations there and reduce the uncertainty at next step. The latter strategy is adopted in this paper. The differential information entropy is a measure of uncertainty. It is optimized to compute the control signal $\mathbf{u}_{i,t}$.

The information entropy of a Gaussian random variable $f_{*i,t}$ conditioned on observation $\mathbf{z}_{N,t}$ is a monotonic function of its variance

$$\begin{aligned} H(f_{*i,t}|\mathbf{z}_{N,t}) &= \frac{1}{2}\log\left(2\pi e \det(\Sigma_{*i,t})\right) \\ &= \frac{1}{2}\log\left(2\pi e \det(K_{**} - Q_{*M_i} C_{i,t}^{-1} Q_{M_i *})\right). \end{aligned}$$

Remark 5: From the above equation, it looks like that the observation is not related to the information entropy. However, the observation affects the hyperparameters because of the online learning algorithm. Thus it is related to the covariance function. As the observation is dependent on position $\mathbf{x}_{*i,t}$, the information entropy can be interpreted as a function of position $\mathbf{x}_{*,t}$, as

$$H(\mathbf{x}_{*i,t}) = H(f_{*i,t}|\mathbf{z}_{N,t}).$$

B. CVT

It is necessary for multiple mobile sensor nodes to move with coordination when they are exploring the area of interest. The CVT approach proposed in [28] is an effective way for motion coordination. The CVT algorithm is a locational optimization approach based on a utility function. Here we propose to use the information entropy $H(\mathbf{x}_{*i,t})$ as the utility function for node i in the CVT algorithm.

A Voronoi tessellation consists of multiple Voronoi cells $V_{i,t}$, each of which is occupied by a sensor node at time step t. A Voronoi cell $V_{i,t}$ is defined as follows:

$$V_{i,t} = \left\{ \mathrm{x}_{*i,t} \in Q \mid \left\| \mathrm{x}_{*i,t} - \mathrm{x}_{i,t} \right\| \le \left\| \mathrm{x}_{*i,t} - \mathrm{x}_{j,t} \right\|, \forall i \ne j \right\}.$$

CVT is a special Voronoi tessellation that requires each sensor node to move forward to the mass center of its Voronoi cell. The utility function is defined as

$$U(X_{N,t}) = \sum_{i=1}^{N} \int_{V_{i,t}} \frac{1}{2} \left\| \mathbf{x}_{*i,t} - \mathbf{x}_{i,t} \right\|^2 H(\mathbf{x}_{*i,t}) d\mathbf{x}_{*i,t}.$$

To compute the mass center of each $V_{i,t}$, the following definitions are used:

$$\begin{aligned} M_{V_{i,t}} &= \int_{V_{i,t}} H(\mathbf{x}_{*i,t}) d\mathbf{x}_{*i,t} \\ L_{V_{i,t}} &= \int_{V_{i,t}} \mathbf{x}_{*i,t} H(\mathbf{x}_{*,t}) d\mathbf{x}_{*i,t} \\ C_{V_{i,t}} &= \frac{L_{V_{i,t}}}{M_{V_{i,t}}}. \end{aligned}$$

The partial derivative of the utility function with respect to position $\mathbf{x}_{i,t}$ is

$$\begin{aligned} \frac{\partial U(X_{N,t})}{\partial \mathbf{x}_{i,t}} &= -\int_{V_{i,t}} (\mathbf{x}_{*i,t} - \mathbf{x}_{i,t}) H(\mathbf{x}_{*i,t}) d\mathbf{x}_{*i,t} \\ &= -M_{V_{i,t}}(C_{V_{i,t}} - \mathrm{x}_{i,t}). \end{aligned}$$

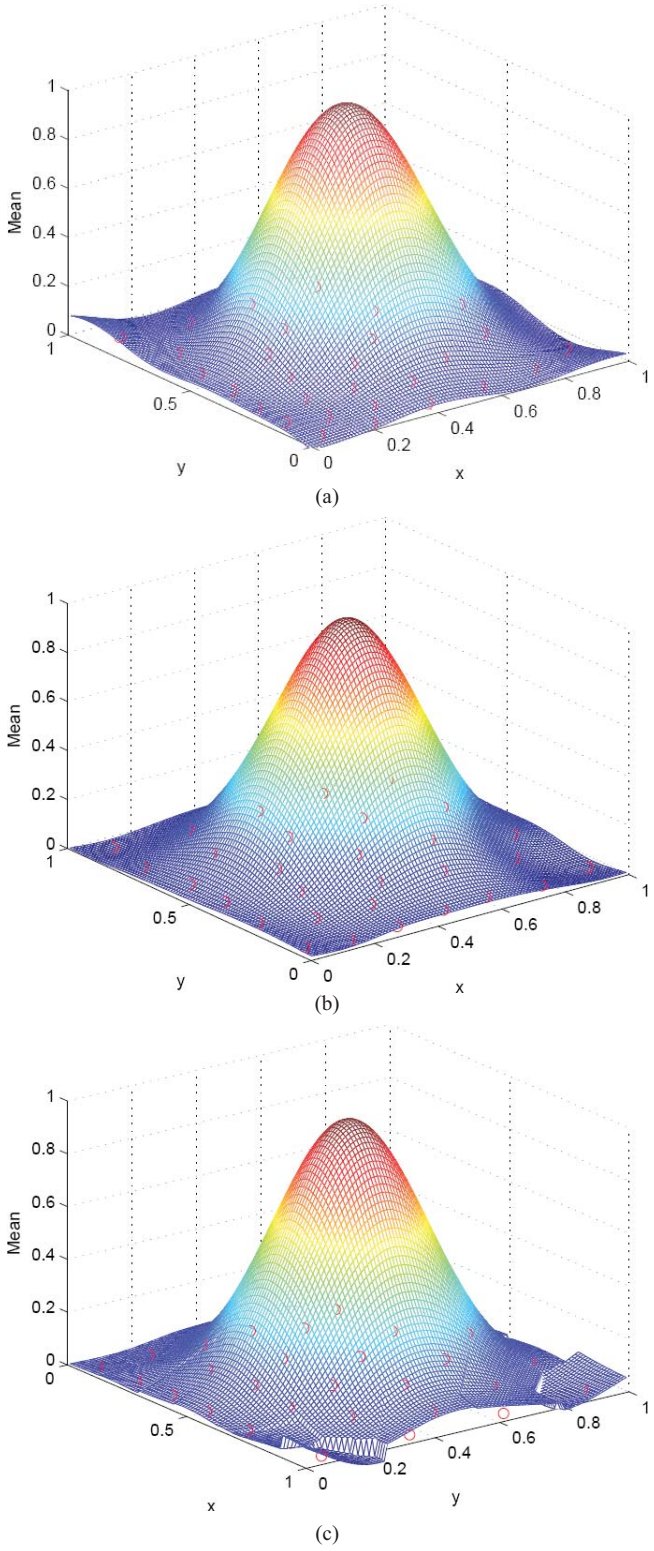


Fig. 3. Predictive mean function recovered by (a) GPR with Gaussian covariance function, (b) DGPR with Wu's compactly supported covariance function, and (c) DGPR with Matern covariance function.

The control input signal of node i is

$$\mathbf{u}_{i,t} = k_u \frac{\partial U(X_{N,t})}{\partial \mathbf{x}_{i,t}}$$

where k_u is the control gain.

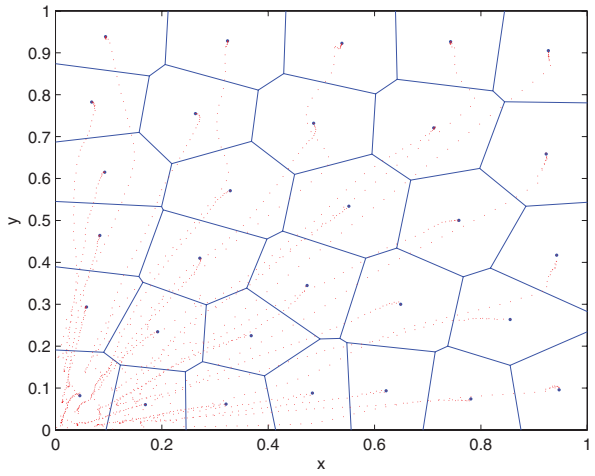


Fig. 4. Mobile sensor trajectories produced by DGPR.

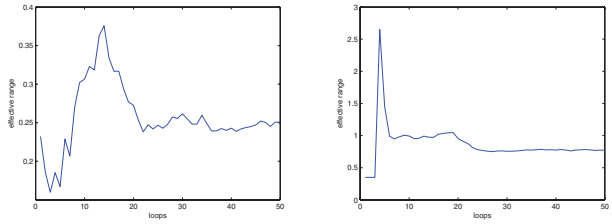


Fig. 5. Effective range of covariance function for (a) GPR and (b) DGPR.

V. Simulations

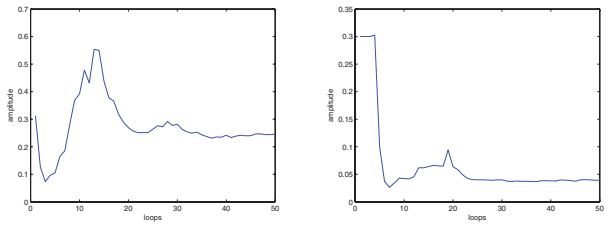
A. Stationary Spatial Functions

A mobile sensor network with $N = 30$ sensor nodes was to be deployed to a 1×1 area to model stationary spatial functions. Initially they were randomly placed in a small area with a size of 0.2×0.2. The hyperparameters to be learned were the effective range $l_{i,t}$ and the amplitude $a_{i,t}$, while the sensor variance $\sigma_{i,t} = 0.01$ was taken as constant. A Gaussian-like spatial function

$$z(x, y) = e^{-\frac{(x-0.5)^2+(y-0.5)^2}{0.07}}$$

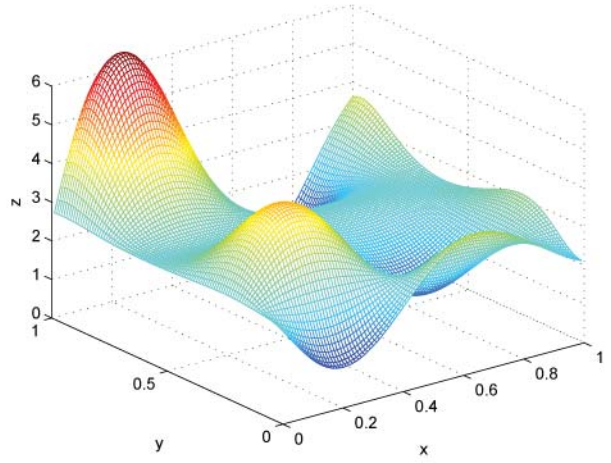
was simulated first by using the GPR with the Gaussian covariance function, then with the DGPR with the Wu's compactly supported covariance function, and, finally, with the DGPR with the Matern covariance function ($\nu = 3/2$). The control loop of the CVT algorithm was selected as 50.

After the simulation, the predictive mean functions are shown in Fig. 3(a) for the GPR with the Gaussian covariance function, Fig. 3(b) for DGPR with the Wu's compactly supported covariance function, and Fig. 3(c) for DGPR with the Matern covariance function ($\nu = 3/2$). Visually, all of them can reconstruct the Gaussian-like function. The result of the DGPR with the Wu's compactly supported covariance function was very close to that of the GPR with the Gaussian covariance function. The result of the DGPR with the Matern covariance function suffered from discontinuities at the corners of the area.

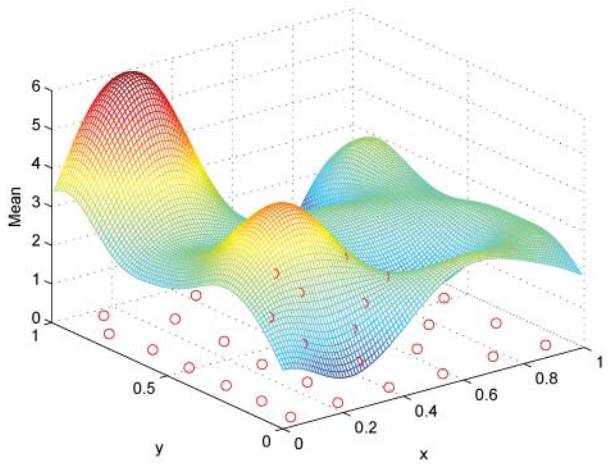


(a) (b)

Fig. 6. Amplitude of the covariance function for (a) GPR and (b) DGPR.



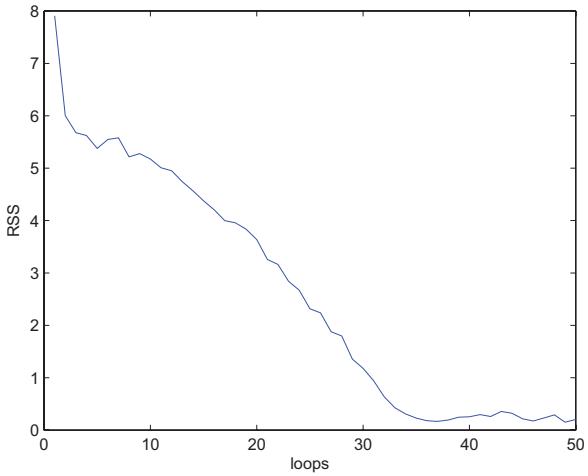
(a)



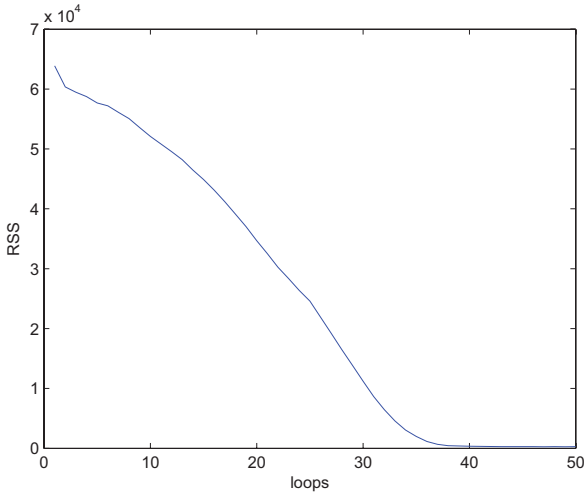
(b)

Fig. 7. (a) Ground-truth function and (b) predictive mean function of DGPR.

Fig. 3(b) and (c) present similar shapes in most of the area except at the corners. The similarity in most of the area confirms the equivalence between the Wu's compactly supported covariance function and the Matern covariance function ($\nu = 3/2$) for regression purpose in this scenario. The differences at the corners of the area are caused by the finite and infinite supported property. Wu's compactly supported covariance function converges to zero at the boundary of communication limit due to the finite supported property, which leads to a smooth result. In contrast, the Matern covariance function does not converge to zero at the



(a)



(b)

Fig. 8. RSS for (a) GPR and (b) DGPR.

boundary of communication limit due to the infinite supported property, which leads to the discontinuities at the corners. The discontinuity can be reflected by a first-order gradient measure, which is defined as the maximum value of the square root of the first-order gradients in the x and y directions

$$\max_{(x,y)\in Q} \sqrt{\left(\frac{\partial \mu}{\partial x}\right)^2 + \left(\frac{\partial \mu}{\partial y}\right)^2}$$

where Q is the area where the mobile sensor network is deployed and μ is the predictive mean function. The maximum value is used here to show the worst case scenario. The gradient measure is 0.0330 and 0.0334 in Fig. 3(a) and (b), respectively. They are very close. However, the gradient measure is 0.0648 in Fig. 3(c). It nearly doubles the values of the smooth regression results and thus reflects the existence of discontinuities.

Mobile sensor trajectories produced by the DGPR with the Wu's compactly supported covariance function are shown in Fig. 4. All the sensor nodes were initially placed at the bottom left corner of the area. They were able to move to cover as large an area as possible. A similar result was obtained from the GPR and hence is omitted here.

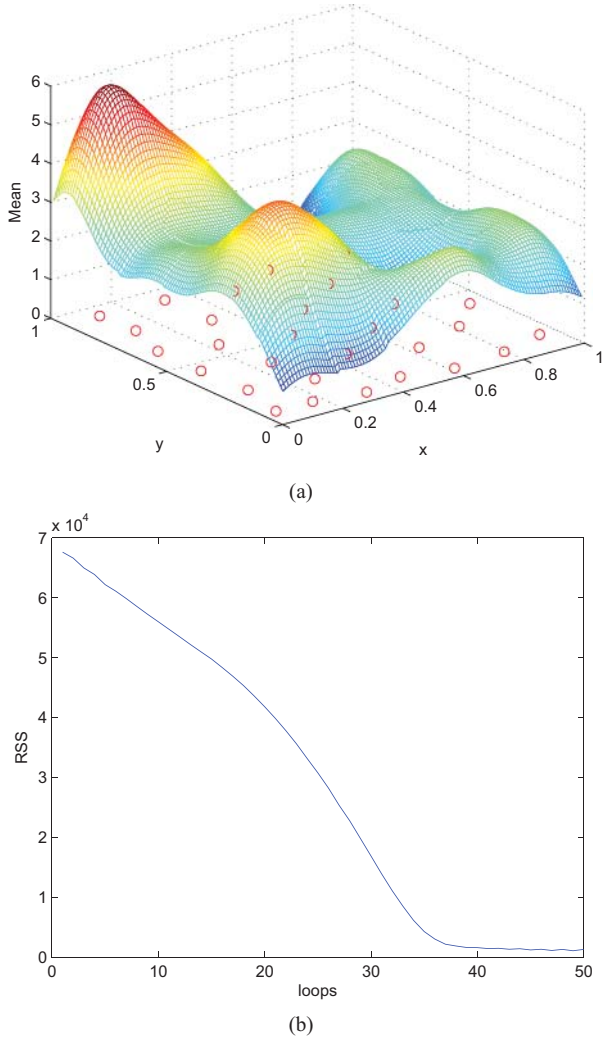


Fig. 9. (a) Predictive mean function. (b) RSS with fixed effective range under DGPR.

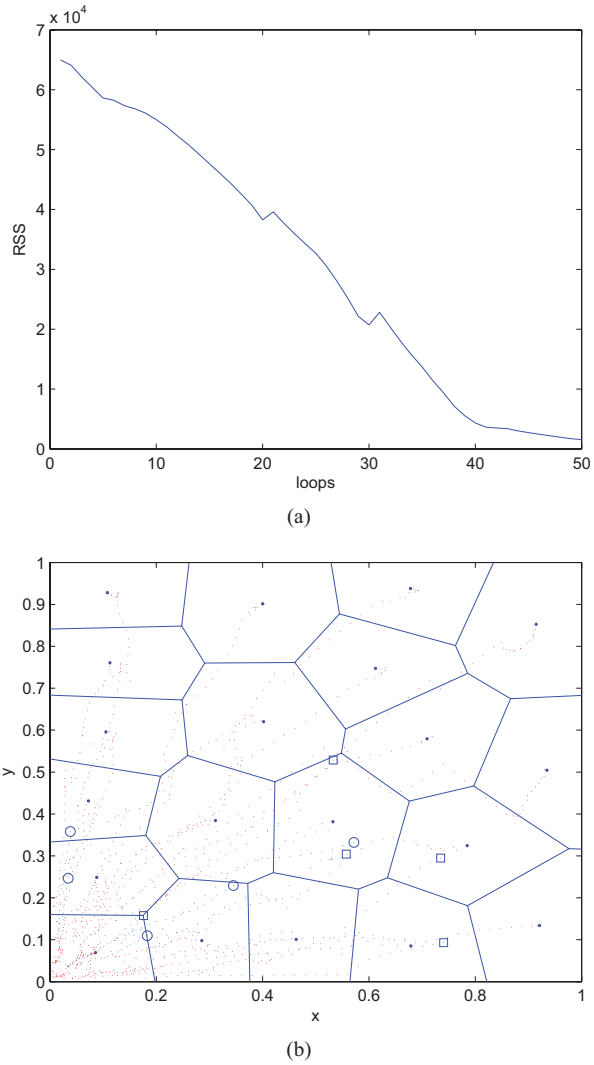


Fig. 10. (a) RSS with faulty nodes. (b) Mobile node trajectories with faulty nodes.

The effective range and the amplitude were learned online using the maximum likelihood algorithm. The learned results are shown in Fig. 5 for the effective range and in Fig. 6 for the amplitude. These parameters experienced some changes at the early stage of the process. After about 25 loops, they gradually reached stable values.

In the second simulation, a complex 2-D spatial function

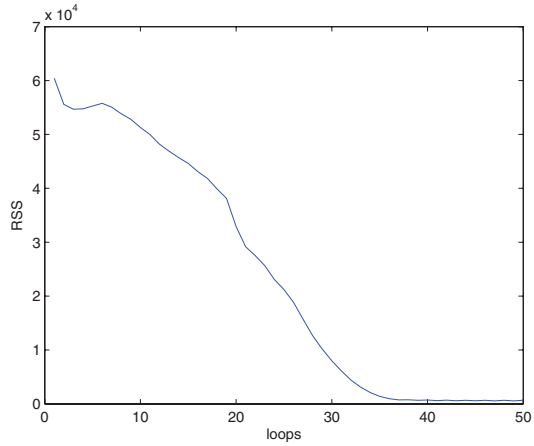
$$z(x, y) = 1.9\big(1.35 + e^{x} \sin\big(13(x - 0.6)^2\big)e^{-y} \sin(7y)\big)$$

was simulated. All the parameters were the same as in the first simulation. The ground truth function is shown Fig. 7(a). In the following presentation, the DGPR is referred to as the DGPR with the Wu's compactly supported covariance function.

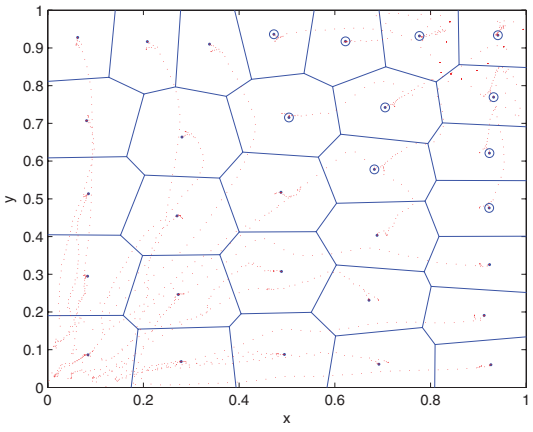
The predictive mean function of DGPR is shown in Fig. 7(b). Visually, their shapes are very similar although there are some errors found at the boundaries. By dividing the area into 100×100 cells, the residual sum of squares (RSS) was calculated at the center of cells at the end of each loop. The results are shown in Fig. 8(a) for GPR and 8(b) for DGPR. Both of them demonstrated an error-reducing behavior and became very small after about 35 loops. However, a less stable behavior of GPR compared to DGPR was observed because of the difference of their covariance functions.

To demonstrate the effect of limited communication on the result, the constraint $l_{i,t} = D_i = 0.4$ was used while the amplitude was still learned. The predictive mean function and the RSS of DGPR are shown in Fig. 9. The shape of the predictive mean function was still similar to the ground-truth function, but there were some minor changes. This is due to the fact that a short effective range was used. The RSS result still had a stable behavior, but it had a larger RSS than the result in Fig. 8(b).

The algorithm robustness and scalability were tested for this 2-D spatial function. First, the algorithm robustness was tested against node failure. All the simulation parameters were the same as before. Five nodes failed at the 20th loop and another five failed at the 30th loop. The RSS result is shown in Fig. 10(a). It can be seen that the RSS value was increased at the 20th and 30th loops due to the failure occurrences, but it still decreased after the interruptions. The circles and squares shown in Fig. 10(b) are the faulty nodes at the 20th and 30th

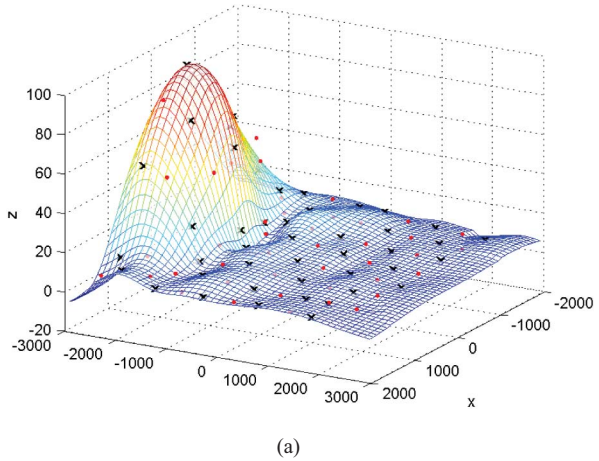


(a)

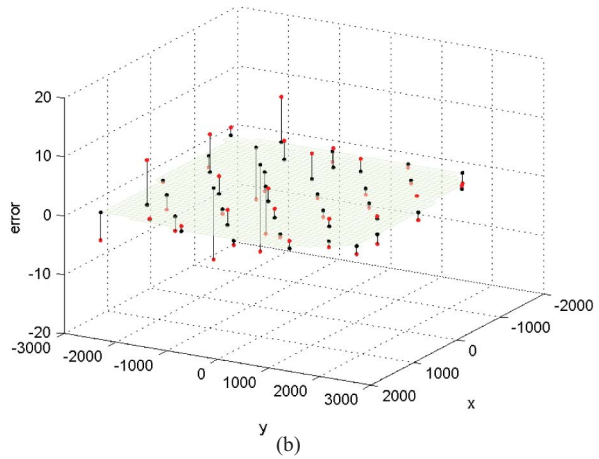


(b)

Fig. 11. (a) RSS with the joining of more nodes. (b) Mobile node trajectories with the joining of more nodes.



(a)



(b)

Fig. 12. (a) DGPR result for light intensity test, including the training samples (blue crosses) and evaluation samples (red dots). (b) Regression errors (red dots) on the evaluation positions (black dots).

loop, respectively. All the other nodes could still successfully cover the area.

Second, the algorithm scalability was tested against the joining of more nodes. Twenty nodes were used initially and they were spread out for 20 loops. At the 20th loop, another 10 nodes located at the top right corner of the area joined the team. All 30 nodes worked together after the 20th loop. The RSS result is shown in Fig. 11(a). The RSS value was reduced at the 20th loop because of the joining of the 10 nodes. Eventually, all the nodes worked as a team to cover the area. Fig. 11(b) shows their trajectories, with the circles representing the nodes joining the team at the 20th loop.

B. Light Intensity Regression

To evaluate the DGPR algorithm in a physical environment, a test on light intensity was conducted. All the parameters of the algorithm were the same as in the previous simulations and the Wu's compactly supported covariance function was used. The environment was an area with the size of 4000 mm by 6000 mm. A light source was placed on one side of the area and the light intensity was sampled by using the light sensor of a LEGO NXT robots. The testing area was also equipped with a 3-D tracking system VICON, which provided the position information of the samples. In total, 88 samples were collected and they were separated into two groups. One group was used to train the DGPR model, and the other group was used to evaluate the prediction result. Both the training and evaluation procedure were conducted offline. The result is shown in Fig. 12(a). The training samples (black crosses) just got stuck on the surface of the regression model, while the evaluation samples (red dots) were slightly away from the surface. The errors (red dots) between the evaluation samples and the model prediction results were calculated and are shown in Fig. 12(b). The evaluation samples (black dots) were evenly distributed in the area. The average absolute error on the evaluation samples was 3.5%. The maximum error for this test was 15.5%. In general, the prediction result fit well with the actual measurement.

C. Spatiotemporal Function

A 2-D Sine–Gordon equation is a spatiotemporal function that has an exact analytic solution [29]. The Sine–Gordon equation has the following expression:

$$\frac{\partial^2 z(t,x,y)}{\partial x^2}+\frac{\partial^2 z(t,x,y)}{\partial y^2}-\frac{\partial^2 z(t,x,y)}{\partial t^2}=\sin(z(t,x,y)).$$

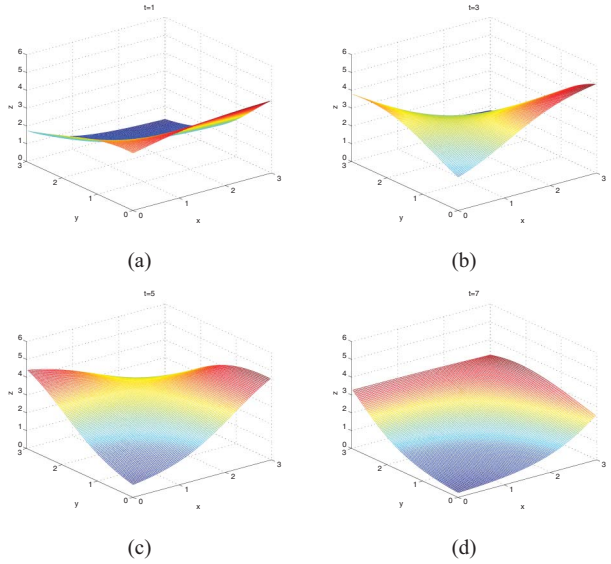


Fig. 13. Sine–Gordon equation (a) $t = 1$, (b) $t = 3$, (c) $t = 5$, and (d) $t = 7$.

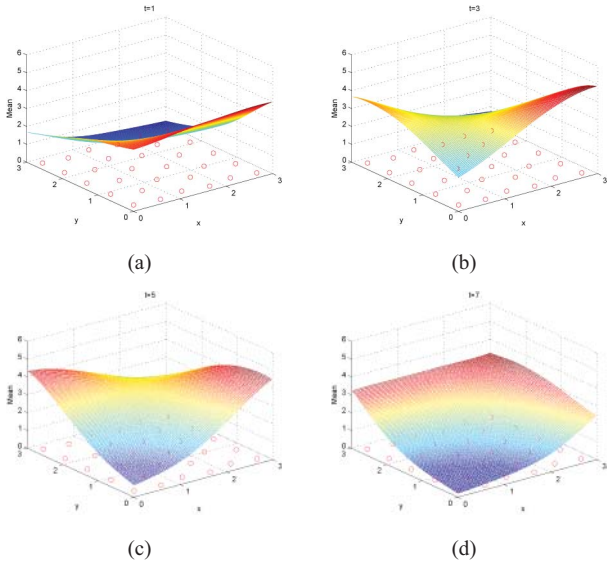


Fig. 14. Predictive mean function reconstructed by DGPR (a) $t = 1$, (b) $t = 3$, (c) $t = 5$, and (d) $t = 7$.

Its solution has the following form:

$$z(t, x, y) = 4 \tan^{-1} \left(\frac{g(t, x, y)}{f(t, x, y)} \right)$$

where

$$\begin{aligned} f(t, x, y) &= 1 + a_{12} e^{\tau_1 + \tau_2} + a_{13} e^{\tau_1 + \tau_3} + a_{23} e^{\tau_2 + \tau_3} \\ g(t, x, y) &= e^{\tau_1} + e^{\tau_2} + e^{\tau_3} + a_{12} a_{13} a_{23} e^{\tau_1 + \tau_2 + \tau_3} \\ a_{ij} &= \frac{(P_i - P_j)^2 + (p_i - p_j)^2 - (w_i - w_j)^2}{(P_i + P_j)^2 + (p_i + p_j)^2 - (w_i + w_j)^2} \\ \tau_i &= P_i x + p_j y - w_i t \end{aligned}$$

provided that the following conditions are satisfied:

$$P_i^2 + p_j^2 - w_i^2 = 1 \quad \text{for } i = 1, 2, 3$$

$$\det \begin{pmatrix} P_1 & p_1 & w_1 \\ P_2 & p_2 & w_2 \\ P_3 & p_3 & w_3 \end{pmatrix} = 0.$$

The parameters used in the simulation were as follows: $P_1 = 1.1$, $P_2 = 0.3$, $P_3 = 0.3$, $p_1 = 0$, $p_2 = 1.2$, $p_3 = 1.2$, $w_1 = 0.4583$, $w_2 = 0.6633$, and $w_3 = 0.6633$. The ground-truth solution with an area of 3×3 is shown in Fig. 13 for $t = 1, 3, 5$, and 7.

A mobile sensor network with $N = 30$ sensor nodes was used. Initially they were randomly placed in the area. The hyperparameters to be learned were the effective range $l_{i,t}$ and the amplitude $a_{i,t}$. The Sine–Gordon equation was tested with 10 time steps starting with $t = 0$ and ending with $t = 9$. Between each two consecutive time steps, DGPR and CVT algorithms were executed iteratively and the number of loops was selected as 5. The predictive mean function is shown in Fig. 14 for $t = 1, 3, 5$, and 7. By comparing with the ground-truth solutions in Fig. 13, it can be seen that the DGPR can reconstruct the Sine–Gordon function very well.

VI. Conclusion

This paper presented a DGPR for mobile sensor networks to model spatial functions. The proposed DGPR was deduced from the projected process approximation the SGPR combined with a compactly supported covariance function. With the proposed DGPR, the hyperparameter learning was also implemented in a distributed manner and the hyperparameters could be updated online. This paper also made use of the advantage of the GPR over other regression techniques for collective motion decision-making. To this end, an information entropy-based CVT approach was used, which allowed the mobile sensor network to reduce the uncertainty of the GP model. The proposed strategies of cooperative modeling and coordinated motion provided satisfactory results for stationary spatial functions and spatiotemporal functions. As our next step, we would like to test our proposed algorithm in more realistic simulations or real experiments.

References

[1] L. F. Merino, J. R. Caballero, J. M. de Dios, and A. O. Ferruz, "A cooperative perception system for multiple UAVs: Application to automatic detection of forest fires," *J. Field Robot.*, vol. 23, no. 3, pp. 165–184, 2006.

[2] C. E. Corrigan, G. C. Roberts, M. V. Ramana, D. Kim, and V. Ramanathan, "Capturing vertical profiles of aerosols and black carbon over the Indian ocean using autonomous unmanned aerial vehicles," *Atmosp. Chem. Phys. Discuss.*, vol. 7, no. 4, pp. 11429–11463, 2007.

[3] N. E. Leonard, D. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. Davis, "Collective motion, sensor networks and ocean sampling," *Proc. IEEE*, vol. 95, no. 1, pp. 48–74, Jan. 2007.

[4] M. Lazaro-Gredilla and A. R. Figueiras-Vidal, "Marginalized neural network mixtures for large-scale regression," *IEEE Trans. Neural Netw.*, vol. 21, no. 8, pp. 1345–1351, Aug. 2010.

[5] G. Skolidis and G. Sanguinetti, "Bayesian multitask classification with Gaussian process priors," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2011–2021, Dec. 2011.

[6] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, and A. Lendasse, "OP-ELM: Optimally pruned extreme learning machine," *IEEE Trans. Neural Netw.*, vol. 21, no. 1, pp. 158–162, Jan. 2010.

[7] C. K. I. Williams and C. E. Rasmussen, "Gaussian processes for regression," in *Advances in Neural Information Processing Systems 8*, D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, Eds. Cambridge, MA: MIT Press, 1996, pp. 514–520.

[8] D. J. C. MacKay, "Introduction to Gaussian processes," in *Neural Networks and Machine Learning*, vol. 168, C. M. Bishop, Ed. Berlin, Germany: Springer-Verlag, 1998, pp. 133–165.

[9] C. E. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. Cambridge MA: MIT Press, 2006.
[10] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *J. Mach. Learn. Res.*, vol. 9, pp. 235–284, Feb. 2008.
[11] R. Stranders, A. Rogers, and N. Jennings, "A decentralized, on-line coordination mechanism for monitoring spatial phenomena with mobile sensors," in *Proc. 2nd Int. Workshop Agent Technol. Sensor Netw.*, Estoril, Portugal, 2008, pp. 1–7.
[12] C. Stachniss, C. Plagemann, and A. J. Lilienthal, "Learning gas distribution models using sparse Gaussian process mixtures," *Auton. Robots*, vol. 26, nos. 2–3, pp. 187–202, 2009.
[13] J. L. Ny and G. Pappas, "On trajectory optimization for active sensing in Gaussian process models," in *Proc. IEEE Conf. Decision Control*, Shanghai, China, Dec. 2009, pp. 6286–6292.
[14] J. Cortes, "Distributed Kriged Kalman filter for spatial estimation," *IEEE Trans. Autom. Control*, vol. 54, no. 12, pp. 2816–2827, Dec. 2009.
[15] J. Choi, J. Lee, and S. Oh, "Swarm intelligence for achieving the global maximum using spatio-temporal Gaussian processes," in *Proc. Amer. Control Conf.*, Seattle, WA, 2008, pp. 1–6.
[16] A. Singh, F. Ramos, H. D. Whyte, and W. J. Kaiser, "Modeling and decision making in spatio-temporal processes for environmental surveillance," in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, AK, May 2010, pp. 5490–5497.
[17] M. Schwager, D. Rus, and J. Slotine, "Decentralized, adaptive coverage control for networked robots," *Int. J. Robot. Res.*, vol. 28, no. 3, pp. 357–375, 2009.
[18] K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman, "Decentralized environmental modeling by mobile sensor networks," *IEEE Trans. Robot.*, vol. 24, no. 3, pp. 710–724, Jun. 2008.
[19] S. Martinez, "Distributed interpolation schemes for field estimation by mobile sensor networks," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, pp. 491–500, Mar. 2010.
[20] J. Quinonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *J. Mach. Learn. Res.*, vol. 6, pp. 1935–1959, Dec. 2005.
[21] L. Csato and M. Opper, "Sparse online Gaussian processes," *Neural Comput.*, vol. 14, no. 2, pp. 641–669, 2002.
[22] M. Seeger, C. K. I. Williams, and N. Lawrence, "Fast forward selection to speed up sparse Gaussian process regression," in *Proc. 9th Int. Workshop Artif. Intell. Stat.*, 2003, pp. 1–8.
[23] Z. Wu, "Compactly supported positive definite radial functions," *Adv. Comput. Math.*, vol. 4, no. 1, pp. 283–292, 1995.
[24] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*. New York: Dover, 1965.
[25] M. L. Stein, *Interpolation of Spatial Data*. New York: Springer-Verlag, 1999.
[26] R. Furrer, M. G. Genton, and D. Nychka, "Covariance tapering for interpolation of large spatial datasets," *J. Comput. Graph. Stat.*, vol. 15, no. 3, pp. 502–523, 2006.
[27] R. Schaback, "Creating surfaces from scattered data using radial basis functions," in *Mathematical Methods in Computer Aided Geometric Design III*, M. Daehlen, T. Lyche, and L. Schumaker, Eds. Nashville, TN: Vanderbilt Univ. Press, 1995, pp. 477–496.
[28] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Auton.*, vol. 20, no. 2, pp. 243–255, Apr. 2004.
[29] L. Guo and S. A. Billings, "State-space reconstruction and spatio-temporal prediction of lattice dynamical systems," *IEEE Trans. Autom. Control*, vol. 52, no. 4, pp. 622–632, Apr. 2007.

Dongbing Gu (SM'07) received the B.Sc. and M.Sc. degrees in control engineering from the Beijing Institute of Technology, Beijing, China, and the Ph.D. degree in robotics from the University of Essex, Essex, U.K.

He was an Academic Visiting Scholar with the Department of Engineering Science, University of Oxford, Oxford, U.K., from October 1996 to October 1997. In 2000, he joined the University of Essex as a Lecturer. Currently, he is a Reader with the School of Computer Science and Electronic Engineering, University of Essex. His current research interests include multiagent systems, wireless sensor networks, distributed control algorithms, distributed information fusion, cooperative control, reinforcement learning, fuzzy logic and neural network-based motion control, and model predictive control.

Huosheng Hu (SM'01) received the M.Sc. degree in industrial automation from Central South University, Beijing, China, and the Ph.D. degree in robotics from the University of Oxford, Oxford, U.K.

He is currently a Professor of computer science with the University of Essex, Essex, U.K., and leads the Human Centered Robotics Group. He has published over 180 research papers in journals, books, and conference proceedings in his areas of expertise. His current research interests include mobile robotics, sensors integration, data fusion, distributed computing, intelligent control, behavior and hybrid control, cooperative robotics, telerobotics, and service robots.

Prof. Hu is currently the Editor-in-Chief of the *International Journal of Automation and Computing*. He is also a reviewer for a number of international journals, such as the IEEE TRANSACTIONS ON ROBOTICS, the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, the IEEE TRANSACTIONS ON NEURAL NETWORKS, and the *International Journal of Robotics Research*. He is a member of IEE, AAAI, the Association of Computing Machinery (ACM), IAS, and IASTED, and is a Chartered Engineer.