

Learning Gas Distribution Models using Sparse Gaussian Process Mixtures

Cyrill Stachniss · Christian Plagemann ·
Achim J. Lilienthal

Received: date / Accepted: date

Abstract In this paper, we consider the problem of learning two-dimensional spatial models of gas distributions. To build models of gas distributions that can be used to accurately predict the gas concentration at query locations is a challenging task due to the chaotic nature of gas dispersal. We formulate this task as a regression problem. To deal with the specific properties of gas distributions, we propose a sparse Gaussian process mixture model, which allows us to accurately represent the smooth background signal and the areas with patches of high concentrations. We furthermore integrate the sparsification of the training data into an EM procedure that we apply for learning the mixture components and the gating function. Our approach has been implemented and tested using datasets recorded with a real mobile robot equipped with an electronic nose. The experiments demonstrate that our technique is well-suited for predicting gas concentrations at new query locations and that it outperforms alternative and previously proposed methods in robotics.

Keywords Gas distribution modeling · gas sensing · Gaussian processes · mixture models

C. Stachniss

University of Freiburg, Dept. of Computer Science, Georges Koehler Allee 79, 79110 Freiburg, Germany
Tel.: +49-761-203-8024, Fax: +49-761-203-8007, E-mail: stachnis@informatik.uni-freiburg.de

C. Plagemann

Stanford University, Computer Science Dept., 353 Serra Mall, Stanford, CA 94305-9010, USA
Phone: +1-650-723-9558, Fax: +1-412-725-1449, E-mail: plagemann@stanford.edu

A. J. Lilienthal

University of Örebro, AASS Research Institute, Fakultetsgatan 1, 70182 Örebro, Sweden
Tel.: +46-19-30-3602, Fax: +46-19-30-3463, E-mail: achim@lilienthals.de

1 Introduction

The problem of modeling gas distributions has important applications in industry, science, and every-day life. Mobile robots equipped with gas sensors can be deployed for pollution monitoring in public areas [DustBot, 2008], surveillance of industrial facilities producing harmful gases, or inspection of contaminated areas within rescue missions.

Although humans have a comparably good odor sensor allowing to distinguish between around 10 000 odors, it is hard for us to build spatial representations of sensed gas distributions. Building gas distribution maps is a challenging task in principle due to the chaotic nature of gas dispersal and because only point measurements of gas concentration are available. The complex interaction of gas with its surroundings is dominated by two physical effects. First, on a comparably large timescale, *diffusion* mixes the gas with the surrounding atmosphere achieving a homogeneous mixture of both in the long run. Second, turbulent air flow fragments the gas emanating from a source into intermittent *patches* of high concentration with steep gradients at their edges [Roberts and Webster, 2002]. This chaotic system of localized patches of gas makes the modeling problem a hard one. In addition, gas sensors provide information about a small spatial region only since gas sensor measurements require direct interaction between the sensor surface and the molecules to be analyzed. This makes gas sensing different to perceiving the environment with other popular robotic sensors like laser range finders, with which a larger area can be measured directly.

Fig. 1 illustrates actual gas concentration measurements recorded with a mobile robot along a corridor containing a single gas source. The distribution consists of a rather smooth “background” signal and several peaks, which indicate high gas concentrations. The challenge in gas distribution mapping is to model this background signal while being able to cover also the areas of high concentration and their sharp boundaries.

From a probabilistic point of view, the task of modeling a gas distribution can be described as finding a model that best explains the observations and that is able to accurately predict new ones. A suitable measure for evaluating models and for comparing alternative ones is to consider the *predictive data likelihood* of an *independent test set*. This measure compares each test data point (which is not contained in the training set) with a predictive distribution estimated by the model. For this, it does not require insight into the model internals and it does not depend on any explicit notion of model complexity or the number of model parameters as, for example, the Bayesian Information Criterion (BIC). The predictive data likelihood is therefore the measure of choice for evaluating especially nonparametric models. As a drawback, one needs a sufficiently large amount of data to be able to separate out a test set without risking that the training set becomes too small to capture the sought-after distribution. The gas mapping application comes with an abundance of available data, such that the predictive test set likelihood constitutes a robust measure for model accuracy.

Simple spatial averaging, which represents a straight-forward approach to the modeling problem, disregards the different nature of the background concentration and the peaks resulting from areas of high gas concentrations and, thus, achieves only limited prediction accuracy. On the other hand, precise physical simulation of the gas dynamics in the environment would require immense computational resources as well as precise knowledge about the physical conditions, which is not known in most practical scenarios.

To achieve a balance between model accuracy and efficiency, we treat gas distribution mapping as a supervised regression problem. We derive a solution by means of a sparse mixture model of Gaussian processes [Tresp, 2000] that is able to handle both physical phenomena highlighted above. Formally, we interpret gas sensor measurements obtained from static sensors or from a mobile robot as noisy samples from a time-constant distribution.

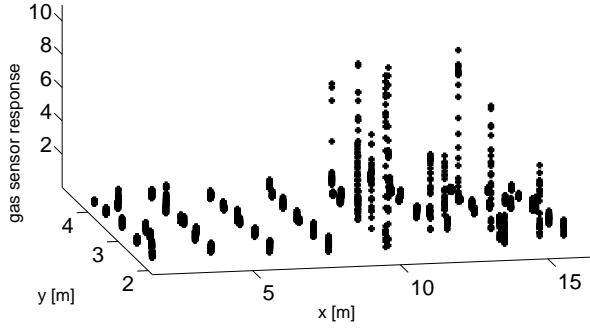


Fig. 1 Gas concentration measurements acquired by a mobile robot in a corridor. The distribution consists of a rather smooth “background” signal and several peaks, which indicate high gas concentrations.

This implies that the gas distribution in fact exhibits a time-constant structure, an assumption that is often made in unventilated and un-populated indoor environments [Wandel *et al.*, 2003].

While existing approaches to gas distribution mapping, such as averaging [Ishida *et al.*, 1998, Purnamadjaja and Russell, 2005, Pyk *et al.*, 2006] or kernel extrapolation [Lilienthal and Duckett, 2004] represent the average concentration per location only, our mixture model actually allows us to do both, computing the mean gas concentration as well as the multi-modal, predictive densities. We further obtain a more accurate estimate of the gas concentration by distinguishing explicitly different components of the distribution, particularly a “background” component where the concentration varies smoothly and a second component that corresponds to the area in which localized patches of gas occur. In a scenario with a constant, uniform airflow, the latter mixture component represents the gas plume [Murlis *et al.*, 1992].

As a by-product, we present a generic algorithm that learns a GP mixture model and at the same time reduces the number of used training data points in order to achieve an efficient representation even for large data sets. We demonstrate in experiments carried out with real mobile robots that our model has a lower mean squared error and a higher data likelihood on test data sets than other existing methods for gas distribution modeling. Thus, it allows to predict gas concentration at query locations more accurately.

This article is organized as follows. After introducing our mixture model in Sec. 2, we propose our method for learning the model components from data and for achieving a sparse approximation in Sec. 3. We then present experimental results involving real mobile platforms in Sec. 4 and discuss related work in Sec. 5.

2 A Mixture Model for Gas Distributions

The general gas distribution mapping problem given a set of concentration measurements $y_{1:n}$ acquired at locations $\mathbf{x}_{1:n}$, is to learn a predictive model $p(y_* | \mathbf{x}_*, \mathbf{x}_{1:n}, y_{1:n})$ for gas concentrations y_* at a query location \mathbf{x}_* . We approach this problem in a nonparametric way, i.e., not assuming a parametric form of the underlying function $f(\cdot)$ in $y = f(\mathbf{x}) + \epsilon$, using the Gaussian process model [Rasmussen and Williams, 2006]. In this Bayesian approach to the non-linear regression problem, one places a prior on the space of functions $p(f)$ using

the following definition: A Gaussian process is a collection of random variables, any of which have a joint Gaussian distribution. More formally, if we assume that $\{(\mathbf{x}_i, f_i)\}_{i=1}^n$ with $f_i = f(\mathbf{x}_i)$ are samples from a Gaussian process and define $\mathbf{f} = (f_1, \dots, f_n)^\top$, we have

$$\mathbf{f} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}), \quad \boldsymbol{\mu} \in \mathbb{R}^n, \mathbf{K} \in \mathbb{R}^{n \times n}. \quad (1)$$

For simplicity of notation, we can assume $\boldsymbol{\mu} = \mathbf{0}$, since the expectation is a linear operator and, thus, for any deterministic mean function $m(\mathbf{x})$, the Gaussian process over $f'(\mathbf{x}) := f(\mathbf{x}) - m(\mathbf{x})$ has zero mean.

The interesting part of the model is indeed the covariance matrix \mathbf{K} . It is specified by $[\mathbf{K}]_{ij} := \text{cov}(f_i, f_j) = k(\mathbf{x}_i, \mathbf{x}_j)$ using a *covariance function* k which defines the covariance of any two function values $\{f_i, f_j\}$ sampled from the process given their input vectors $\{\mathbf{x}_i, \mathbf{x}_j\}$ as parameters. Intuitively, the covariance function specifies how *similar* two function values $f(\mathbf{x}_i)$ and $f(\mathbf{x}_j)$ are depending only on the corresponding inputs. The standard choice for k is the squared exponential covariance function

$$k_{SE}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp\left(-\frac{1}{2} \frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{\ell^2}\right), \quad (2)$$

where the so-called *length-scale* parameter ℓ defines the global smoothness of the function f and σ_f^2 denotes the amplitude (or signal variance) parameter. These parameters, along with the global noise variance σ_n^2 that is assumed for the noise component, are known as the *hyperparameters* of the process. They are denoted as $\boldsymbol{\theta} = \langle \sigma_f, \ell, \sigma_n \rangle$.

Given a set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ of training data where $\mathbf{x}_i \in \mathbb{R}^d$ are the inputs and $y_i \in \mathbb{R}$ the targets, the goal in regression is to predict target values $y_* \in \mathbb{R}$ at a new input point \mathbf{x}_* . Let $\mathbf{X} = [\mathbf{x}_1; \dots; \mathbf{x}_n]^\top$ be the $n \times d$ matrix of the inputs and \mathbf{X}_* be defined analogously for multiple test data points. In the GP model, any finite set of samples is jointly Gaussian distributed

$$\begin{bmatrix} \mathbf{y} \\ f(\mathbf{X}_*) \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} k(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I} & k(\mathbf{X}, \mathbf{X}_*) \\ k(\mathbf{X}_*, \mathbf{X}) & k(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix}\right), \quad (3)$$

where $k(\mathbf{X}, \mathbf{X})$ refers to the covariance matrix built by evaluating the covariance function $k(\cdot, \cdot)$ for all pairs of all row vectors $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ of \mathbf{X} . To make predictions at \mathbf{X}_* , we obtain the predictive mean

$$\bar{f}(\mathbf{X}_*) := \mathbb{E}[f(\mathbf{X}_*)] = k(\mathbf{X}_*, \mathbf{X}) \left[k(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I} \right]^{-1} \mathbf{y} \quad (4)$$

and the (noise-free) predictive variance

$$\mathbb{V}[f(\mathbf{X}_*)] = k(\mathbf{X}_*, \mathbf{X}_*) - k(\mathbf{X}_*, \mathbf{X}) \left[k(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I} \right]^{-1} k(\mathbf{X}, \mathbf{X}_*), \quad (5)$$

where \mathbf{I} is the identity matrix. The corresponding (noisy) predictive variance for an observation \mathbf{y}_* can be obtained by adding the noise term σ_n^2 to the individual components of $\mathbb{V}[f(\mathbf{X}_*)]$.

The standard GP model recapitulated above has two major limitations in our problem domain. First, the computational complexity is high, since to compute the predictive variance given in Eq. (5), one needs to invert the matrix $k(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}$. This introduces a complexity of $\mathcal{O}(n^3)$ where n is the number of training examples. As a result, an important

issue for GP-based solutions to practical problems is the reduction of this complexity. This can, as we will show in Sec. 3, be achieved by artificially limiting the training data set in a way that introduces small loss in the data likelihood while at the same time minimizing the runtime. As a second limitation, the standard GP model generates a *uni-modal* distribution per input location \mathbf{x} . This assumption hardly fits our application domain in which a relatively smooth “background” signal is typically *mixed* with medium- and high-concentration “packets” of gas. In the following, we address this issue by deriving a *mixture model* of Gaussian processes.

2.1 Mixtures of Gaussian Process Models

The GP mixture model [Tresp, 2000] constitutes a locally weighted sum of several Gaussian process models. For simplicity of notation, we consider without loss of generality the case of single predictions only (\mathbf{x}_* instead of \mathbf{X}_*). Let $\{\mathcal{GP}_1, \dots, \mathcal{GP}_m\}$ be a set of m Gaussian processes representing the individual mixture components. Let $P(z(\mathbf{x}_*) = i)$ be the probability that \mathbf{x}_* is associated with the i -th component of the mixture. Let $\bar{f}_i(\mathbf{x}_*)$ be the mean prediction of \mathcal{GP}_i at \mathbf{x}_* . The likelihood of observing y_* is thus given by

$$h(\mathbf{x}_*) := p(y_* | \mathbf{x}_*) = \sum_{i=1}^m P(z(\mathbf{x}_*) = i) \mathcal{N}_i(y_*; \mathbf{x}_*), \quad (6)$$

where we define $\mathcal{N}_i(y; \mathbf{x})$ as the Gaussian density function with mean $\bar{f}_i(\mathbf{x})$ and variance $\mathbb{V}[f_i(\mathbf{x})] + \sigma_n^2$ evaluated at y . One can sample from such a mixture by first sampling the mixture component according to $P(z(\mathbf{x}_*) = i)$ and then sampling from the corresponding Gaussian. For some applications such as information-driven exploration missions, it is practical to estimate the mean and variance for this multi-modal model. The mean $\mathbb{E}[h(\mathbf{x}_*)]$ of the mixture model is given by

$$\bar{h}(\mathbf{x}_*) := \mathbb{E}[h(\mathbf{x}_*)] = \sum_{i=1}^m P(z(\mathbf{x}_*) = i) \bar{f}_i(\mathbf{x}_*) \quad (7)$$

and the corresponding variance is computed as

$$\mathbb{V}[h(\mathbf{x}_*)] = \sum_{i=1}^m P(z(\mathbf{x}_*) = i) \left(\mathbb{V}[f_i(\mathbf{x}_*)] + (\bar{f}_i(\mathbf{x}_*) - \bar{h}(\mathbf{x}_*))^2 \right). \quad (8)$$

2.2 The Choice of the Covariance Function

The covariance function in a Gaussian Process as well as in our mixture model is a crucial component as it encodes knowledge about the function to approximate. It specifies the dependency between two function values $f(\mathbf{x}_i)$, $f(\mathbf{x}_j)$ and this dependency is computed only based on the corresponding inputs.

The standard choice for covariance function is the squared exponential (SE) shown in Eq. (2), however, there are several other possibilities to define a covariance function. In this paper, we also analyze how the choice of the covariance function affects the quality of the gas distribution model. In detail, we analyze the squared exponential and two instances of the Matérn covariance function.

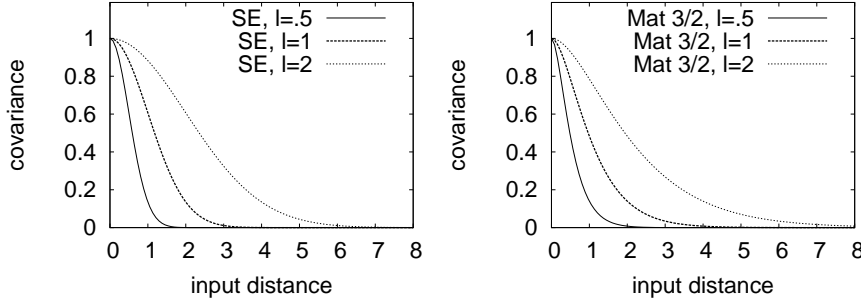


Fig. 2 Example plots of the squared exponential covariance function (left) and the Matérn 3/2 covariance function (right), each plotted for varying hyperparameters.

In case of the Matérn covariance function, we consider the so-called “Matérn 3/2” and “Matérn 5/2” functions among the class of Matérn kernels. They are given by

$$k_{Mat\ 3/2}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \left(1 + \frac{\sqrt{3} \|\mathbf{x}_i, \mathbf{x}_j\|}{l} \right) \exp \left(-\frac{\sqrt{3} \|\mathbf{x}_i, \mathbf{x}_j\|}{l} \right) \quad (9)$$

and

$$k_{Mat\ 5/2}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \left(1 + \frac{\sqrt{5} \|\mathbf{x}_i, \mathbf{x}_j\|}{l} + \frac{5 \|\mathbf{x}_i, \mathbf{x}_j\|^2}{3l^2} \right) \exp \left(-\frac{\sqrt{5} \|\mathbf{x}_i, \mathbf{x}_j\|}{l} \right). \quad (10)$$

As for the case of the SE covariance function, the parameter ℓ is the length-scale that defines the global smoothness of the function f and σ_f^2 denotes the amplitude (or signal variance) parameter.

Fig. 2 shows 2d-plots of these covariance functions illustrating the assumed dependency between data points of the function to model depending only on the distance of the inputs.

When comparing the properties of the individual covariance functions, the SE function is a rather smooth one and, therefore, leads to a comparably strong smoothing of the function approximation. This, however, might contradict the nature of gas distribution as well as other physical phenomena. Therefore, we also consider the Matérn covariance function that typically produces rougher estimates and thus might be better suited for the problem studied in this paper. Among the two Matérn kernels used in this paper, the Matérn 5/2 is smoother than the Matérn 3/2.

3 Learning the Model from Data

Given a training set $\mathcal{D} = \{(\mathbf{x}_j, y_j)\}_{j=1}^n$ of gas concentration measurements y_j and the corresponding sensing locations \mathbf{x}_j , the task is to jointly learn the assignment $z(\mathbf{x}_j)$ of data points to mixture components and, given this assignment, the individual regression models \mathcal{GP}_i . Tresp [2000] describes an approach based on Expectation Maximization (EM) for solving this task. We take his approach, but also seek to minimize the number of training data points to achieve a computationally tractable model even for large training data sets \mathcal{D} . This is of major importance in our application, since typical gas concentration data sets easily exceed $n = 1\,000$ data points and the standard GP model (see Sec. 2) is of cubic complexity $\mathcal{O}(n^3)$.

Different solutions have been proposed for lowering this upper bound, such as dividing the input space into different regions and solving these problems individually or by deriving sparse approximations for the whole space. Sparse GPs [Smola and Bartlett, 2000, Snelson and Ghahramani, 2006a] use a reduced set of inputs to approximate the full GP model. This new set can be either a subset of the original inputs [Smola and Bartlett, 2000] or a set of r new pseudo-inputs [Snelson and Ghahramani, 2006a] which are obtained using an optimization procedure. This reduces the complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(nr^2)$ with $r \ll n$, which in practice results in a nearly linear complexity.

We apply a method similar to sparse GPs and select a subset of the original inputs. In the remainder of this section, we describe a greedy forward-selection algorithm integrated into the EM-learning procedure which achieves a sparse mixture model by selecting a subset of the original inputs, while also maximizing the cross validation data likelihood.

3.1 Initializing the Mixture Components

In a first step, we subsample n_1 data points and learn a standard GP for this set (including the optimization of the hyperparameters). This model \mathcal{GP}_1 constitutes the first mixture component. To improve the estimate of gas concentration in areas that are poorly modeled by this initial model, we learn an “error model”, termed \mathcal{GP}_Δ , that captures the absolute differences between a set of target values and the predictions of \mathcal{GP}_1 . We then sample points according to \mathcal{GP}_Δ and use them to initialize the next mixture component. In this way, the new mixture is initialized with the data points that are poorly approximated by the first one and a hyperparameter optimization is performed. This process is repeated until the desired number of model components is reached. For typical gas modeling scenarios, we found that two mixture components are sufficient to achieve good results. In our experiments, the converged mixture models nicely reflect the bimodal nature of gas distributions, having one smooth “background” component and a layer of locally concentrated structures.

It should be mentioned, that depending on the actual data, the error model (“error GP”) might have to be evaluated at all $n - n_1$ inputs which would lead to large computational overhead. Instead, we actually average multiple spatially close measurements and evaluate only at uniformly sampled locations. This is clearly an approximation but only used for the error model of our approach. We, however, did not encounter problems using this strategy which is actually used only for initialization.

3.2 Iterative Learning via Expectation-Maximization

The Expectation Maximization (EM) algorithm can be used to obtain a maximum likelihood estimate when hidden and observable variables need to be estimated. It consists of two steps, the so-called estimation (E) step and the maximization (M) step which are executed alternately.

In the E-step, we estimate the probability $P(z(\mathbf{x}_j) = i)$ that data point j corresponds to model component i . This is done by computing the likelihood of each data point for the model components individually. Thus, the new $P(z(\mathbf{x}_j) = i)$ is computed given the previous estimate as

$$P(z(\mathbf{x}_j) = i) \leftarrow \frac{P(z(\mathbf{x}_j) = i) \mathcal{N}_i(y_j; \mathbf{x}_j)}{\sum_{k=1}^m P(z(\mathbf{x}_j) = k) \mathcal{N}_k(y_j; \mathbf{x}_j)} . \quad (11)$$

In the M-step, we update the components of our mixture model. This is achieved by integrating the probability that a data point belongs to a model component into the individual GP learning steps (see also [Tresp, 2000]). This is achieved by modifying Eq. (4) to

$$\bar{f}_i(\mathbf{X}_*) = k(\mathbf{X}_*, \mathbf{X}) \left[k(\mathbf{X}, \mathbf{X}) + \Psi^i \right]^{-1} \mathbf{y}, \quad (12)$$

where Ψ^i is a matrix with

$$[\Psi^i]_{jj} = \frac{\sigma_n^2}{P(z(\mathbf{x}_j) = i)} \quad (13)$$

and zeros in the off-diagonal elements. Eq. (5) is updated accordingly. The matrix Ψ^i allows us to consider the probabilities that the individual inputs belong to the corresponding components. Figuratively speaking, the contribution of an unlikely data point to a model is reduced by increasing the data point specific noise term. If the assignment probability, on the other hand, is one, only σ_n^2 remains and the point is fully included as in the standard GP model.

Learning a GP model also involves the estimation of its hyperparameters $\theta = \{\sigma_f, \ell, \sigma_n\}$. To estimate them for \mathcal{GP}_i , we first apply a variant of the hyperparameter heuristic used by Snelson and Ghahramani [2006a] in their open-source implementation. We extended it to incorporate the correspondence probability $P(z(\mathbf{x}_k) = i)$ into this initial guess

$$\ell \leftarrow \max_{\mathbf{x}_j} P(z(\mathbf{x}_j) = i) \|\mathbf{x}_j - \bar{\mathbf{x}}\| \quad (14)$$

$$\sigma_f^2 \leftarrow \frac{\sum_{j=1}^n P(z(\mathbf{x}_j) = i) (y_j - \mathbb{E}[y])^2}{\sum_{j=1}^n P(z(\mathbf{x}_j) = i)} \quad (15)$$

$$\sigma_n^2 \leftarrow \frac{1}{4} \sigma_f^2, \quad (16)$$

where $\bar{\mathbf{x}}$ refers to the weighted mean of the inputs, each \mathbf{x}_j having a weight of $P(z(\mathbf{x}_j) = i)$. To optimize the hyperparameters based on this initial estimate, one could apply, for example, Rasmussen’s conjugate-gradient-based approach [Rasmussen, 2006]. In our experiments, however, this approach lead to overfitting problems and we therefore resorted to cross validation-based optimization. Concretely, we repeatedly sample hyperparameters and evaluate the model accuracy according to Sec. 3.2 on a separate validation set. As a hyperparameter sampling strategy, we draw in each even iteration of this sampling new hyperparameters from an uninformed prior and in each odd iteration, we improve the current best parameters θ' by sampling from a Gaussian with mean θ' . The standard deviation of that Gaussian is decreased with the iteration number.

In our experiments, this rather straight forward strategy converged quickly after a few iterations (approx. 50 iterations, see Fig. 11 for an example). Note that there are more sophisticated strategies, for example simulated annealing, that can be used instead. However, we selected a simpler approach since it provided satisfactory results and can be implemented with five lines of code.

3.3 Learning the Gating Function

In our mixture model, the gating function defines for each data point the probability of being assigned to the individual mixture components. The EM algorithm learns the assignment

probabilities for the used training inputs \mathbf{x}_j , maximizing the cross validation data likelihood. To generalize these assignments to the whole input space (to form a proper gating *function*), we place another GP prior on the gating variables. Concretely, we learn a gating GP for each component i that uses the \mathbf{x}_j as inputs and the $z(\mathbf{x}_j)$ obtained from the EM algorithm as targets. Let $\bar{f}_i^z(\mathbf{x})$ be the prediction of z for \mathcal{GP}_i . Given this set of m GPs, we can compute the correspondence probability for a new test point \mathbf{x}_* as

$$P(z(\mathbf{x}_*) = i) = \frac{\exp(\bar{f}_i^z(\mathbf{x}_*))}{\sum_{j=1}^m \exp(\bar{f}_j^z(\mathbf{x}_*))} . \quad (17)$$

3.4 Summary

This section briefly summarizes our approach for learning the GP mixture model. First, we initialize the mixture components which are individual GPs. This done by randomly sampling data point for the first component. Then, an error GP is learned to estimate the prediction error. The data points for the subsequent component is then sampled based on the error GP. Second, the we apply the expectation maximization algorithm to optimize the mixture components and to estimate the hidden mixture/class assignment variables. In each iteration of the EM, the hyperparameters for the mixture components are iteratively optimized. Finally, the gating function is learned using again the GP framework. The gating function models the class assignments for the whole input space. Learning is done based on separated training and test sets.

3.5 Illustrating Example

To visualize our approach, we now give a simple, one-dimensional example. The left diagram of Fig. 3 shows simulated data points, of which most were sampled uniformly from the interval $[2 : 2.5]$ and some are distributed with a larger spread at two distinct locations. The same diagram also shows a standard GP model learned on this set, which is not able to explain the data well. The right diagram of the figure shows \mathcal{GP}_Δ , i.e. the resulting error model, which characterizes the local deviations of the model predictions from the data points. Based on this model, a second mixture component is initialized and used as input to the EM algorithm.

The individual diagrams in Fig. 4 illustrate the iterations of the EM algorithm (to be read from left to right and from top to bottom). They depict the two components of the mixture model. The learned gating function after convergence of the algorithm is depicted in the left diagram of Fig. 5. The right diagram in the same figure gives the final GP mixture model. It is clearly visible that the mixture model better represents this data set than the standard GP model, which assumes a smooth, uni-modal process (see the left diagram of Fig. 3).

4 Experimental Results

We carried out pollution monitoring experiments in a real-world setting, in which a mobile robot followed a predefined sweeping trajectory covering the area of interest. Along its path, the robot was stopped for several seconds, 10 s (outdoors) and 30 s (indoors), at predefined points to acquire measurements. The spacing between the grid points was set to values

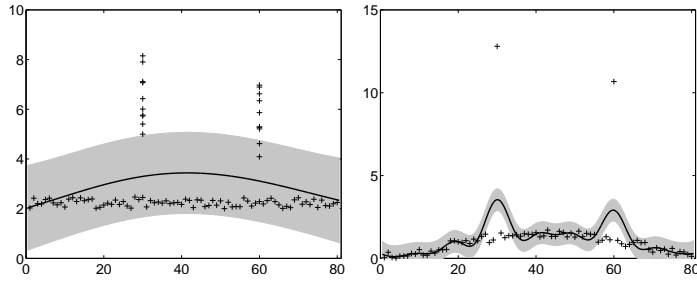


Fig. 3 Left: The standard GP used to initialize the first mixture component. Right: The error GP used to initialize the next mixture component.

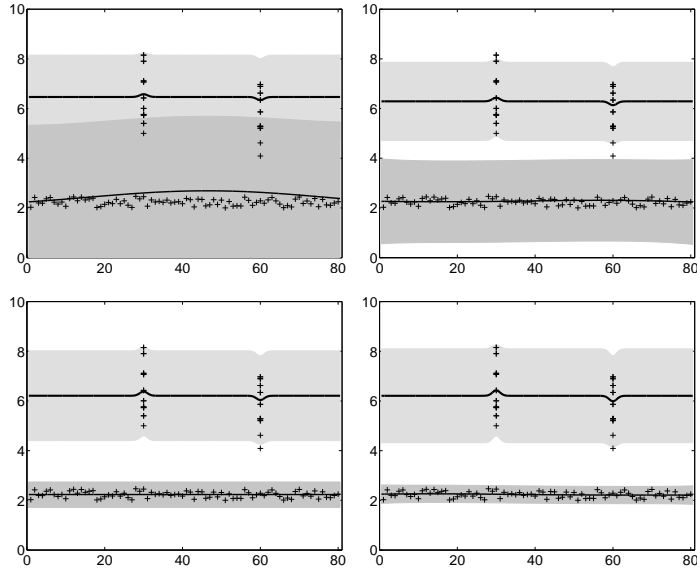


Fig. 4 Components during different iterations of the EM algorithm.

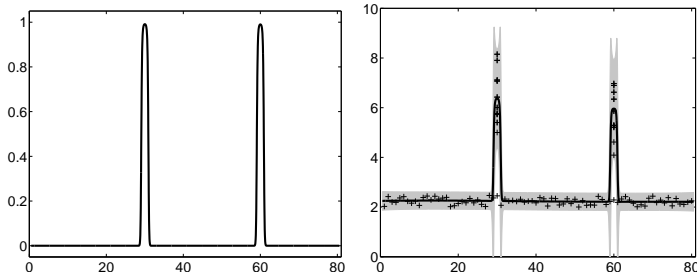


Fig. 5 Left: The learned gating function. Right: Resulting distribution of the GP mixture model.

Dataset	GP (SE)	GP (Mat3/2)	GP (Mat5/2)	GPM avg (SE)	GPM avg (Mat3/2)	GPM avg (Mat5/2)
<i>3-rooms</i>	-1.22	-1.25	-1.27	-1.50	-1.51	-1.52
<i>corridor</i>	-0.98	-1.06	-0.98	-1.58	-1.58	-1.60
<i>outdoor</i>	-1.11	-1.17	-1.22	-1.72	-1.88	-1.85

Table 1 Average negative log likelihoods of test data points for different approaches. The results of the comparison between the GP and the GP mixture model with corresponding covariance functions shown in this table differ significantly (10 repetitions, $\alpha = 5\%$).

Dataset	GP	GPM avg	GPM
<i>3-rooms</i>	-1.22	-1.50	-1.54
<i>corridor</i>	-0.98	-1.58	-1.60
<i>outdoor</i>	-1.11	-1.72	-1.80

Table 2 Comparison between standard GP (GP), the GP mixture model with averaging (GPM avg) according to Eq. (8) and Eq. (7), and the GP mixture model with multi-modal estimates (GPM) based on 10 repetitions (here using the SE covariance function).

between 0.5 m to 2.0 m depending on the topology of the available space, see Fig. 6. In the experiments, the sweeping motion was performed twice in opposite directions which allows us to use the second visit for evaluating our predictions. Due to the slow response of the gas sensors and in order to avoid disturbance to the gas distribution created by the robot itself, the robot was driven at a maximum speed of 5 cm/s in between the stops. The gas source was a small cup filled with ethanol and in the experiments, the robot approached the cup up to a distance of approximately 0.1 m.

The robot was equipped with a SICK laser range scanner used for pose correction, with an electronic nose, and an anemometer. The electronic nose is a Figaro TGS 2620 gas sensor enclosed in an aluminum tube. This tube was mounted horizontally at the front side of the robot. The electronic nose is actively ventilated through a fan that creates a constant airflow towards the gas sensor. This lowers the effect of external airflow and the movement of the robot on the sensor response.

Note that in this work, we concentrate only on the gas concentration measurements and do not consider the pose uncertainty of the vehicle. One can apply one of the various SLAM systems available to account for the uncertainty in the robot’s pose [Frese, 2006, Grisetti *et al.*, 2007, Lilienthal *et al.*, 2007].

4.1 Inspected Environments

Three environments with different properties were selected for the pollution monitoring experiments. The first experiment, termed *3-rooms*, was carried out in an enclosed indoor area that consists of three rooms which are separated by slightly protruding walls in between them. The area covered by the robot is approximately 14 m \times 6 m. There is little exchange of air with the “outer world” in this environment. The gas source was placed in the central room and all three rooms were monitored by the robot. The second location was a part of a *corridor* with open ends and a high ceiling. The area covered by the trajectory of the robot is approximately 14 m \times 2 m. The gas source was placed on the floor in the middle of the investigated corridor segment. Finally, an *outdoor* scenario was considered. Here, the experiments were carried out in an 8 m \times 8 m region that is part of a much bigger open area.

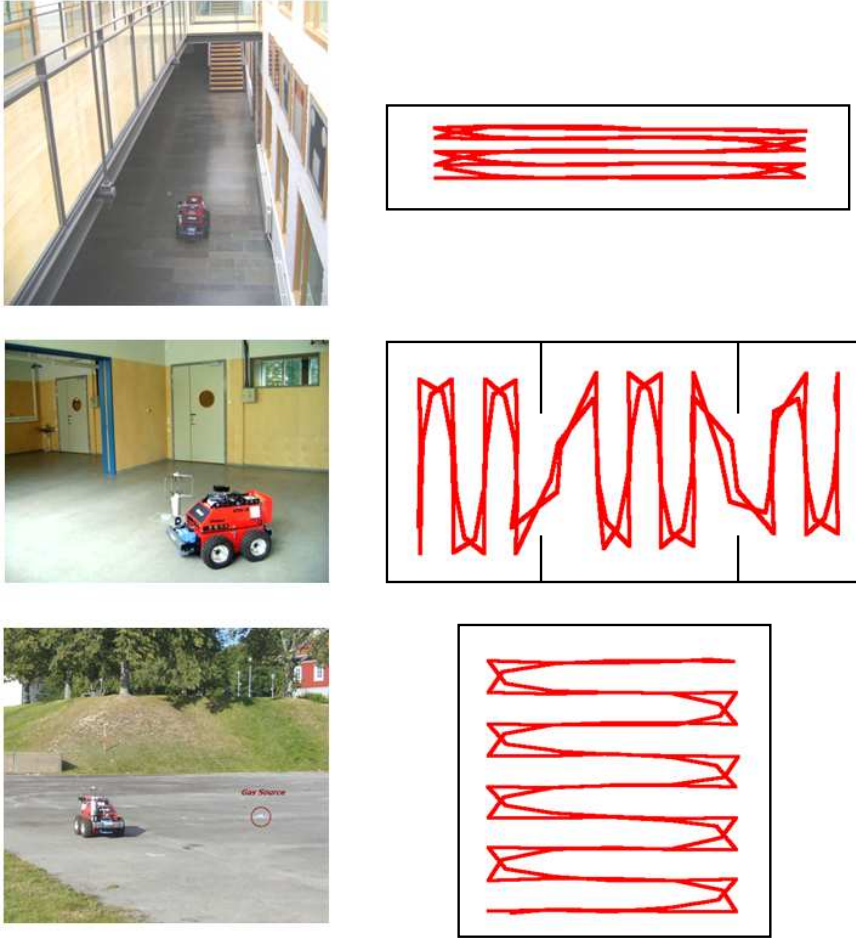


Fig. 6 Pictures of the robot inspecting three different environments as well as the corresponding sweeping trajectories.

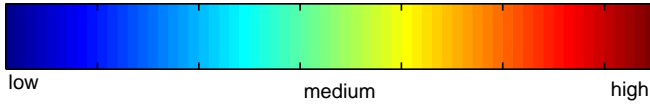


Fig. 7 Color schema for the gas concentrations visualizations (Matlab default). Gas distribution measurements are always normalized between 0 and 1 given the current set of observations used for learning the model.

We used the raw sensor readings in all three environments as training sets and applied our approach to learn the gas distribution models. The robot moved through the environment twice. We used the first run for learning the model and the second one for evaluating it. To benchmark our results, we compare against gas distribution models learned using (a) standard GP regression, (b) a grid-based interpolation approach, and (c) kernel extrapolation. For the Gaussian process regression, we furthermore analyze the influence of different co-

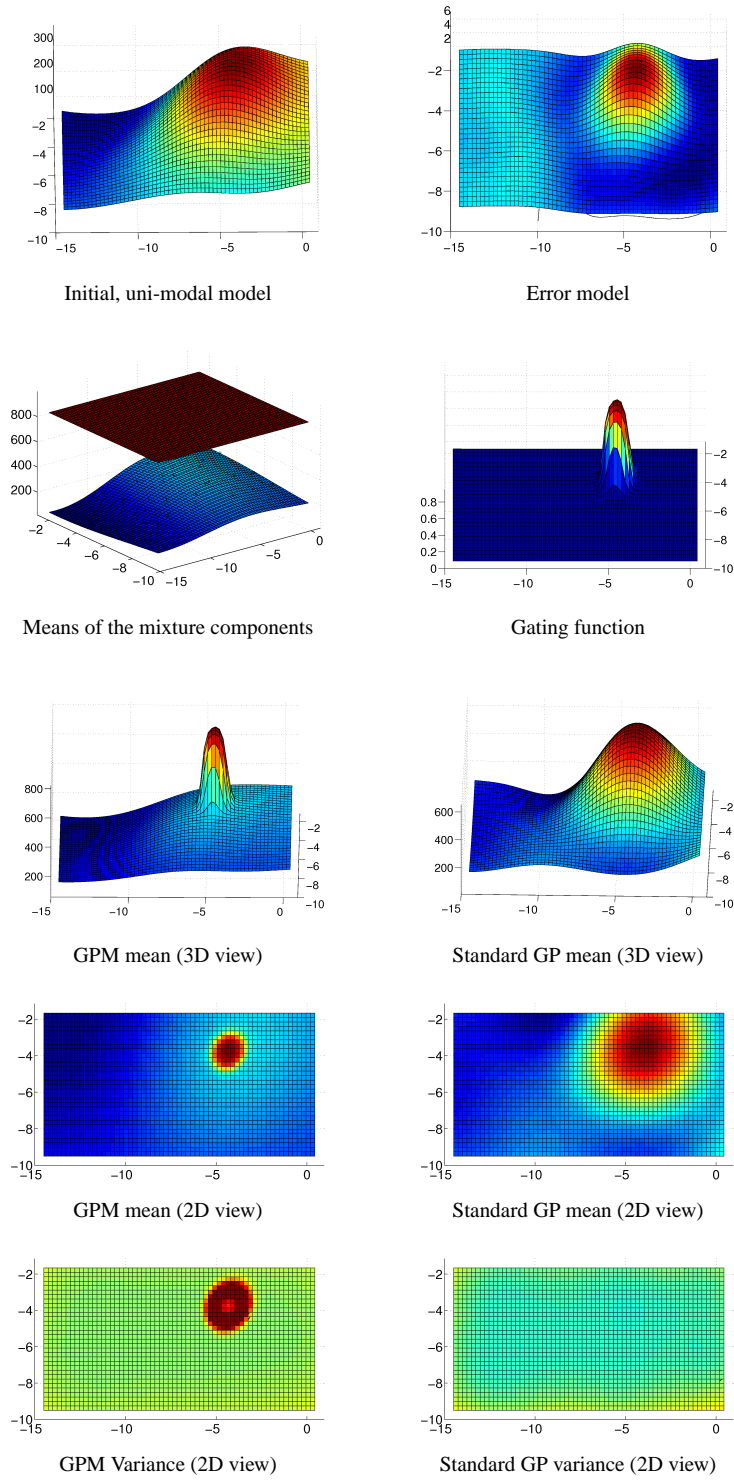


Fig. 8 The 3-rooms dataset with one ethanol gas source in the central room. The room structure itself is not visualized here. In all plots, blue represents low, yellow reflects medium, and red refers to high values. See Fig. 7 for the color encoding. The unit of the x- and y-axis is meter.

variance functions in the obtained results. For the visualizations, we always used the default Matlab color scheme depicted also in Fig. 7 and normalized the gas concentration measurements obtained to values between zero and one.

4.2 Evaluation

Fig. 8 shows the learned models for the *3-room* dataset. The left plot in the first row illustrates the mean prediction for the standard GP on the subsampled training set that defines the first mixture component. The right diagram depicts the error GP representing the differences between the initial prediction and a set of observations. Based on the error GP, a new mixture component is initialized and the EM algorithm is carried out. The means of the two mixture components after convergence are shown in the left diagram of the second row and the learned gating function is visualized in the adjacent diagram on the right. The left diagram in the third row shows the mean prediction of the final mixture model. As can be seen, the model consists of a smooth “background” distribution and a peak of gas concentration—close to the gas source—with a sharp boundary. In contrast to this, the standard GP (right diagram in the third row) learned using the same data is overly smooth for this dataset, especially in proximity to the gas source. For both models, the squared exponential covariance function has been used here.

Table 1 summarizes the negative log likelihoods of the test data (second part of the dataset, which was not used for training) given our mixture model (GPM) as well as the standard GP model (GP). As can be seen, our GPM method outperforms the standard GP model in all settings. A t-test on 10 repeated learning runs revealed that these results are significant ($\alpha = 5\%$). Two reasons for the increased model accuracy of GPM w.r.t. standard GPs can be seen in the 2D plots in the last two rows of Fig. 8. First, as already mentioned before, the standard GP overly smoothes the area close to the gas source and, second, its variance estimates around the source are too low (since standard GPs assume a constant noise rate for the whole domain). The table furthermore analyses the results obtained with different covariance functions. The Matérn kernels perform on average slightly better than the squared exponential function. This is probably the case because the Matérn kernels are less smooth which is in line with the nature of the problem addressed in this paper. In Table 2, we provide two likelihoods for our model, the one given in Eq. (6) (called “GPM” in the table) and the one computed based on the averaged prediction specified in Eq. (7) and Eq. (8) (called “GPM avg”).

Fig. 9 visualizes the final results for the *corridor* experiment for the GPM model (means of the mixture components in the left diagram and the predictive uncertainty on the right). The raw dataset from this experiment is plotted in Fig. 1. In this experiment, the area of high gas concentration was also mapped comparably accurate by the standard GP, but again the variance close to the area of high gas concentration was too small. This can be seen by comparing the images in the right column of Fig. 9, which show the standard GP results for different covariance functions in the top three rows and for the GPM below.

By carefully inspecting the results (best viewed in color), one can see slight differences resulting from the covariance functions. The squared exponential function yields smoother results than the Matérn kernels which can be seen on the border around the areas of high concentrations. The results measured by means of the NLPD computed based on separated test sets over multiple runs illustrate that the GPM models always outperformed the standard model (see tables). Furthermore, the Matérn kernels seem to be slightly better suited to

model gas distributions since they are less smooth compared to the squared exponential function.

Similar results are also obtained in the *outdoor* dataset. Mean and variance predictions of the different GP mixture models with different covariance functions are provided in Fig. 10. The corresponding result of the standard GP including a plot that illustrates the evolution of the negative log likelihood (NLPD) during sampling of the hyperparameters for the standard GP model and mixture GP model (SE covariance) is given in Fig. 11.

In all our experiments, we limited the number of data points in the reduced input set to $n_1 = 100$. The datasets itself contained between 2 500 and 3 500 measurements so our model was able to make accurate predictions with less than 5% of the data. Matrices of that size can be easily inverted. As a result the overall computation time for learning our model including cross validation is around 1 minute for all datasets shown above running Matlab on a standard laptop computer without explicitly optimized code.

Finally, we compared the mean estimates of our mixture model to the results obtained with the method of Lilienthal and Duckett [2004] as well as with an often used approach that uses a grid in combination with linear interpolation like in [Pyk *et al.*, 2006]. The results of this comparison in terms of the MSE measure are shown in Fig. 12. As can be seen from the diagram, our method outperforms both alternative methods.

4.3 Distribution Modeling in an Easy Setup

We also tested our gas distribution modeling algorithm with a “smoother” data set. The electronic nose on the mobile robot is also equipped with a temperature sensor and we used the temperature measurements as input to the gas distribution modeling algorithm proposed in this paper. Even so, the obtained measurements were temperature measurements instead of gas concentration measurements, our approach can be directly applied.

The measurements were recorded along a random sweeping trajectory in a corridor. The data set indicates a roughly linear gradient in the temperature distribution. In this situation, we expect that our mixture model should perform similar compared to the standard GP approach or the kernel extrapolation technique since the simpler techniques are also well suited to model such a function.

We therefore carried out the modeling task based on the temperature datasets with the different approaches. Our expectation was actually matched perfectly in this setting. Both mixture components of our method actually converged to approximatively the same solution and this model is more or less identical to the one generated by the standard GP approach as well as to the kernel extrapolation method. All three approaches yield nearly identical results differing by less than 1%. This holds for the MSE as well as for the NLPD (for GP and GPM), see Fig. 13. Obviously, the standard GP has a lower computational load than the mixture approach and thus is preferable if the designer of the system can ensure that no mixture components are needed to model the data.

5 Related Work

A common approach to creating representations for time-averaged concentration fields is to acquire measurements using a fixed grid of gas sensors over a long period of time. Equidistant gas sensor locations can be used to directly measure and map the average concentration values according to a given grid approximation of the environment. This approach was taken

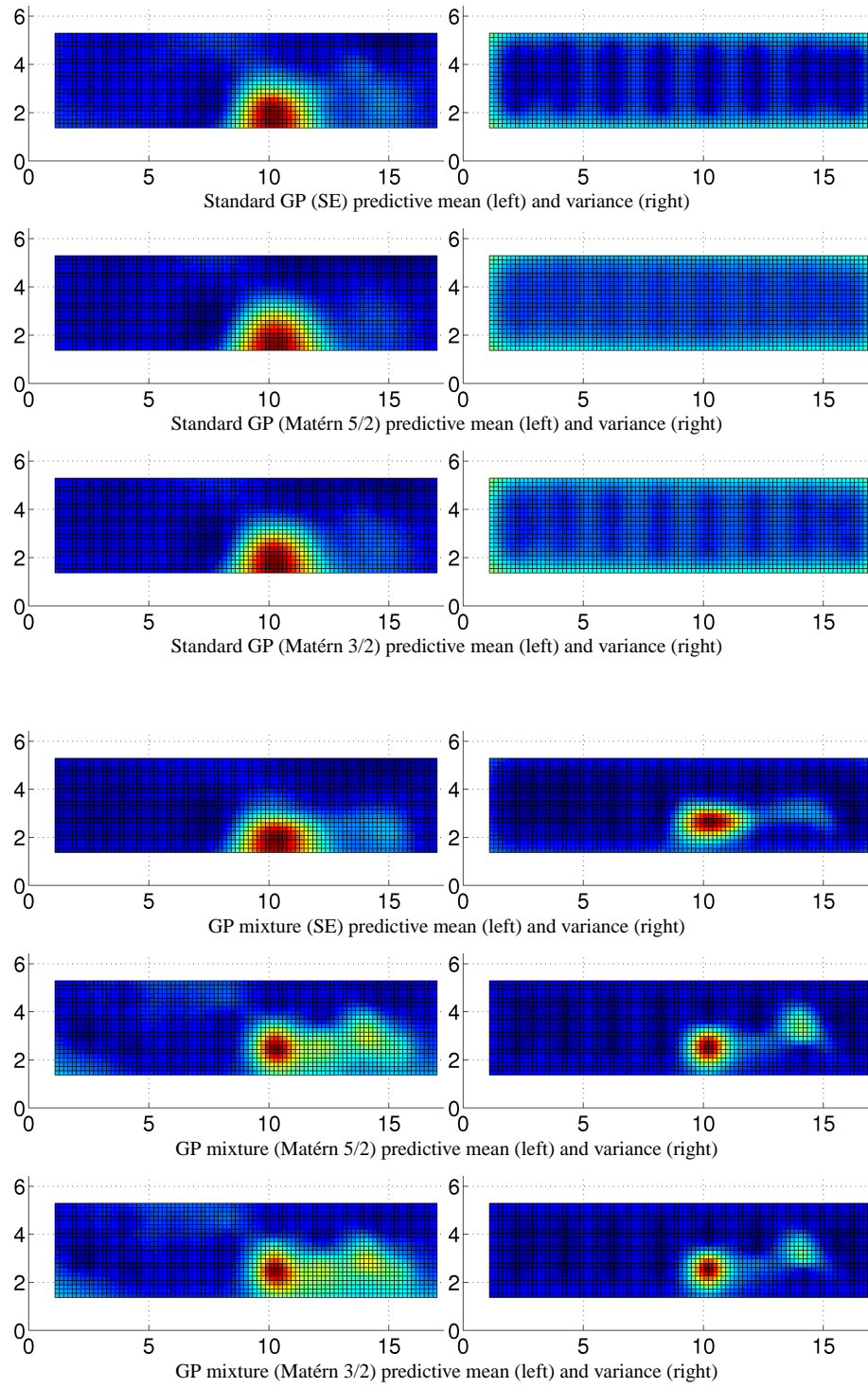


Fig. 9 Models learned from concentration data recorded in the *corridor* environment. The gas source was placed at the location (10, 3). We evaluated the standard GP and the our mixture model all using the different covariance functions. The unit of the x- and y-axis is meter.

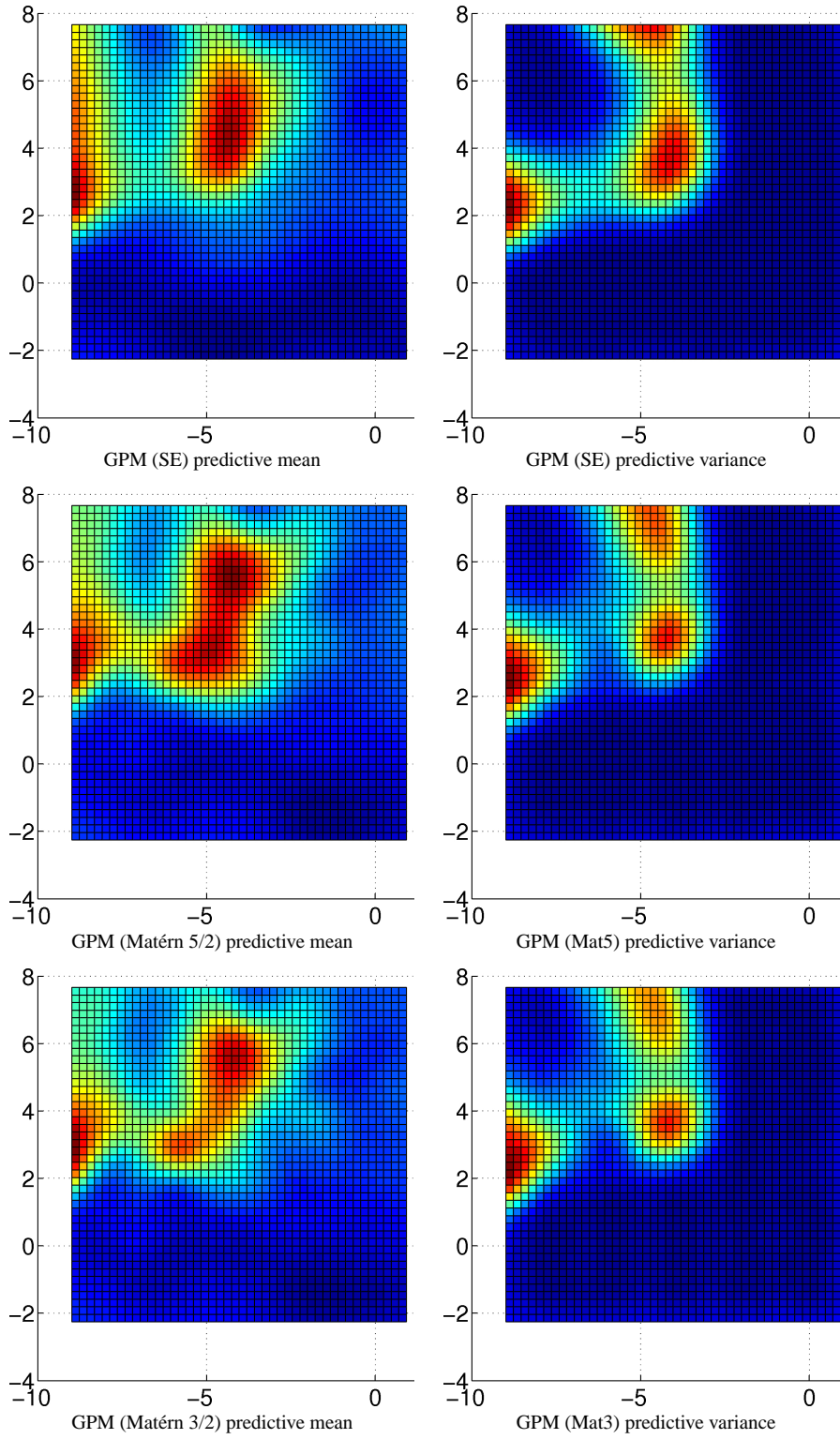


Fig. 10 Results for the *outdoor* dataset in an 8 m by 8 m area with an ethanol source in the center. The measured airflow indicates a major wind direction approximately from south-east to north-west. The unit of the x- and y-axis is meter.

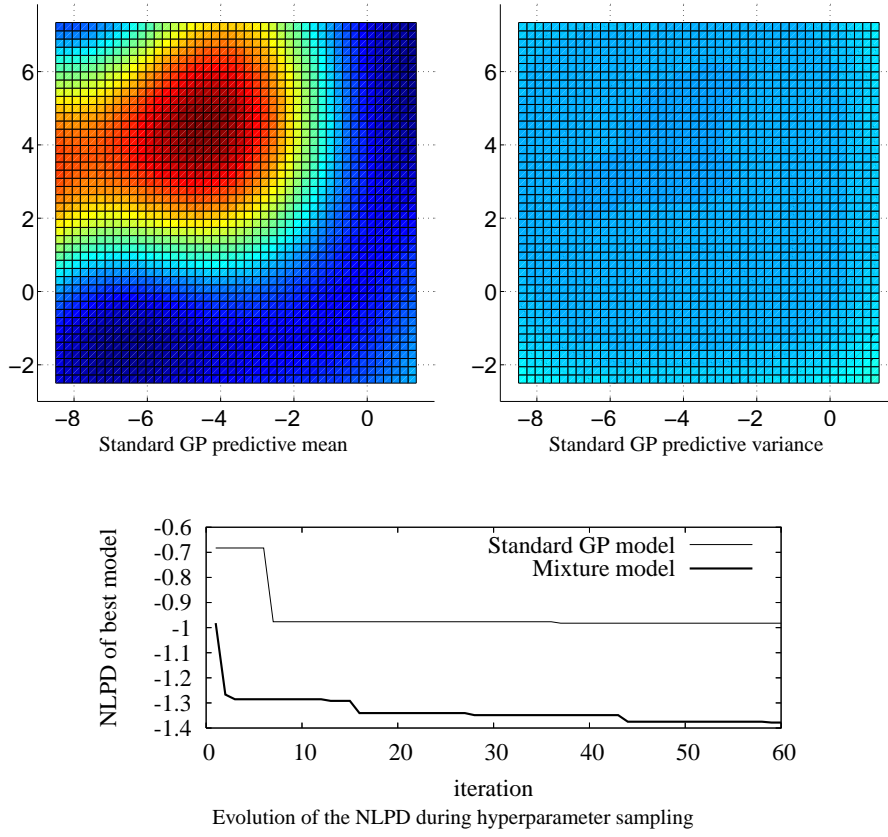


Fig. 11 Corresponding results for the *outdoor* dataset obtained with the standard GP model (top) and the evolution of the NLPD shown for the first 60 iterations (bottom). The unit of the x- and y-axis in the plots in the first row is meter.

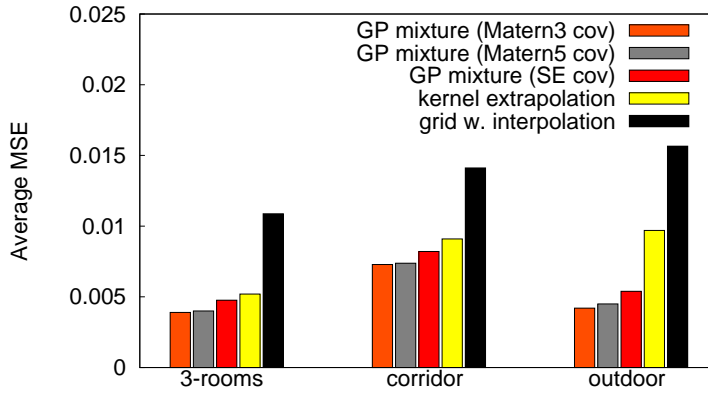


Fig. 12 Experimental comparison of our GP mixture model with different covariance functions to two alternative techniques in three real-world setting. The bars show the mean squared error of predicted compared to the measured concentration on a test set, averaged over 10 runs.

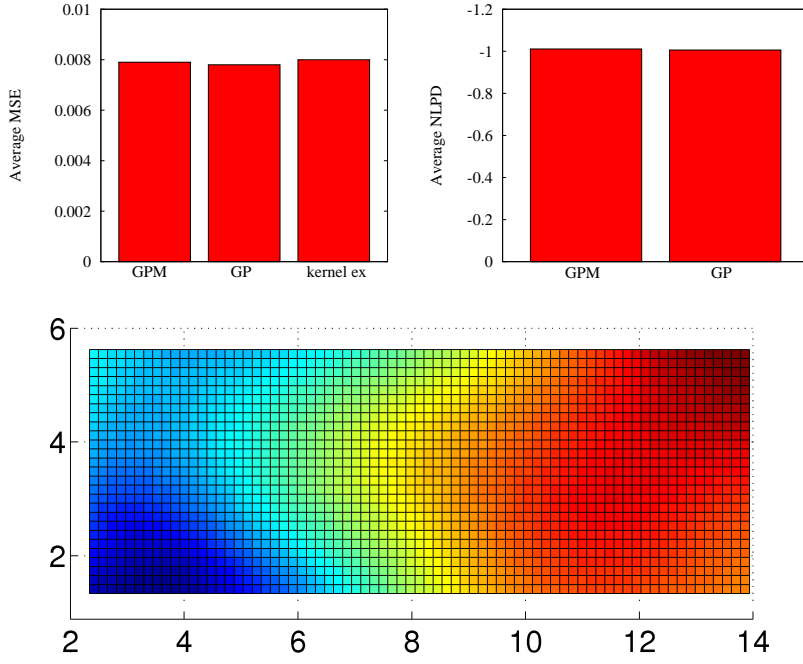


Fig. 13 Experimental comparison of the GP mixture model (GPM), the standard GP model (GP), and kernel extrapolation in a simple setting. Top: As expected, all three approaches perform more or less equal. Bottom: Model learned by the GPM approach (all approaches produced highly similar estimates). The unit of the x- and y-axis in the plots in the bottom row is meter.

by Ishida *et al.* [1998]—additionally considering partially simultaneous measurements. A similar method was used in [Purnamadjaja and Russell, 2005], but instead of the average concentration, the *peak* concentration observed during a sampling period of 20 s was considered to create the map.

Consecutive measurements with a single sensor and time-averaging over 2 minutes for each sensor location were used by Pyk *et al.* [2006] to create a map of the distribution of ethanol. Methods, which aim at determining a map of the instantaneous gas distribution from successive concentration measurements, rely on the assumption of a time-constant distribution profile, i.e. on uniform delivery and removal of the gas to be analyzed as well as on stable environmental conditions. Thus, the experiments of Pyk *et al.* were performed in a wind tunnel with a constant airflow and a homogeneous gas source. To make predictions about locations outside of the directly observed regions, the same authors apply bi-cubic interpolation in the case of equidistant measurements and triangle-based cubic filtering in the case, in which the measurement points are not equally distributed. A problem with such interpolation methods is that there is no means of “averaging out” instantaneous response fluctuations at measurement locations. Even if response values are measured close to each other, they will appear independently in the gas distribution map with interpolated values in between. Consequently, interpolation-based maps tend to become more and more jagged the more new measurements are added [Lilienthal *et al.*, 2006].

Histogram-based methods approximate the continuous distribution of gas concentration by means of binning according to regular grids. Hayes *et al.* [2002] for instance suggest

using two-dimensional histograms over the number of “odor hits” received in the corresponding area. “Odor hits” are counted whenever the response level of a gas sensor exceeds a defined threshold. In addition to the dependency of the gas distribution map on the selected threshold, a disadvantage of processing binary information only is that useful information contained in the (continuous) sensor readings is discarded. Further disadvantages of histogram-based methods for gas distribution modeling are their dependency on a properly chosen bin size and the lack of generalization across bins or beyond the inspection area.

Gas distribution mapping based on kernel extrapolation can be seen as an extension of the histogram-based approach. The idea was introduced by Lilienthal and Duckett [2004]. In this model, spatial integration is carried out by convolving sensor readings and modeling the information content of the point measurements with a Gaussian kernel. As discussed in [Lilienthal *et al.*, 2006], this method is related to nonparametric estimation using Parzen windows. The complexity of model-free approaches for converging to a stable representation—either in terms of time consumption or the number of sensors—scales quadratically with the size of the environment.

A model-based approach to estimate concentration maps has been described by Marques *et al.* [2005]. In this approach, the work space is discretized into a 2-d regular grid and the concentration in each cell is represented by a state variable. Using an advection-diffusion model of chemical transport, a reduced order Kalman filter is applied in order to estimate the state variables corresponding to the grid cells. According to the assumption of a non-turbulent transport model, the experimental run presented was carried out in an indoor environment with artificially introduced laminar airflow of approx. 1.5 m/s. Model-based approaches have also been applied to infer the parameters of an analytical gas distribution model from the measurements [Ishida *et al.*, 1998]. They naturally depend on the characteristics of the assumed model. Complex numerical models based on the simulation of fluid dynamics are computationally expensive and require accurate knowledge of the state of the environment (boundary conditions) which are typically not available in practice. Simpler analytical models, on the other hand, often make rather unrealistic model assumptions which hardly fit the real situation. Model-based approaches also rely on well-calibrated gas sensors and an established understanding of the sensor-environment interaction.

The Kalman filter approach by Marques *et al.* [2005] provides an estimate of the predictive uncertainty. A related approach is the work by Blanco *et al.* [2009] in which a Kalman filter is used for sequential Bayesian estimation on a 2-d grid. Instead of the advection-diffusion model, a stationary distribution is assumed in the latter work. It is important to note that the covariance obtained from these two approaches is the covariance of the mean, which can only decrease as new observations are processed. Since the predictive variance computed with the algorithm proposed in this paper can adapt to the real variability of the measurements at each location, its performance in terms of the average negative log likelihood is substantially better than with the approach by Blanco *et al.* [2009] (personal communication). We believe that this is also true for the mapping algorithm by Marques *et al.* [2005] although the two methods cannot be compared directly due to the strong assumptions on the environmental conditions by Marques *et al.*

In contrast to the above-mentioned approaches, we apply a Gaussian process-based mixture model to the problem of learning probabilistic gas distribution maps. The history of the idea behind the Gaussian process approach to regression dates back to Wiener [1964], Kolmogoroff [1941], O’Hagan [1978], and others (see [Rasmussen and Williams, 2006, Sec. 2.8]). For a detailed and quantitative comparison of GPs with alternative approaches such as neural networks, we refer to [Rasmussen, 1996]. GPs allow us to model the dependencies between measurements by means of a covariance function. They enable us to make

predictions at locations not observed so far and do not only provide the mean gas distribution but also the predictive uncertainty. Our mixture model is furthermore able to model sharp boundaries around areas of high gas concentration. Technically, we build on Tresp’s mixture model of GP experts (see [Tresp, 2000]) better deal with the varying properties in the data. Extensions of this technique using infinite mixtures have been proposed by Rasmussen and Ghahramani [2002] and Meeds and Osindero [2006]. Other model extensions that aim at increasing the expressiveness of Gaussian processes include, e.g., heteroscedastic GPs for modeling input-dependent noise [Le *et al.*, 2005, Kersting *et al.*, 2007, Snelson and Ghahramani, 2006b], nonstationary GPs for modeling input-dependent smoothness [Paciorek and Schervish, 2003, Plagemann *et al.*, 2008, Schmidt and O’Hagan, 2003], or special covariance functions for non-vectorial inputs [Driessens *et al.*, 2006, Collins and Duffy, 2002]. Compared to the latter extensions to the standard GP model, the mixture model approach can be seen as the natural choice for the gas-mapping task, since the distribution of data points is multi-modal. Future work, however, could include a quantitative comparison of the alternative approaches or aim at integrating several of them.

The work presented here extends our previous RSS’2008 paper [Stachniss *et al.*, 2008]. First, we investigated the use of different covariance functions in the GP model for gas distribution mapping. This showed that there are better choices than the previously used squared exponential covariance function. Second, we extended the experimental section providing a larger set of experiments. We furthermore identified and evaluated a scenario which is well designed for the standard GP approach and evaluated the performance of our proposed mixture model. It turned out that in such a situations, designed for the standard GP, our approach performs equally well.

6 Conclusion

We considered the problem of modeling gas distributions from sensor measurements by means of sparse Gaussian process mixture models. Gaussian processes are an attractive modeling technique in this context since they do not only provide a gas concentration estimate for each point in the space but also the predictive uncertainty. Our approach learns a GP mixture model and simultaneously decreases the computational complexity by reducing the training set in order to achieve an efficient representation even for a large number of observations. The mixture model allows us to explicitly distinguish the different components of the spatial gas distribution, namely areas of high gas concentration from the smoothly varying background signal. This improves the accuracy of the gas concentration prediction.

Our method has been implemented and tested using gas sensors mounted on a real robot. With our method, we obtain gas distribution models that better explain the sensor data compared to techniques such as the standard GP regression for gas distribution mapping. Our approach and the one of Lilienthal and Duckett [2004] provide similar mean gas concentration estimates, their approach as well as the majority of techniques in the field, however, lack the ability of also estimating the corresponding predictive uncertainties.

Acknowledgments

This work has partly been supported by the DFG under contract number SFB/TR-8, and by the EC under contract number FP6-045299-Dustbot: Networked and Cooperating Robots

for Urban Hygiene, and FP7-224318-DIADEM: Distributed Information Acquisition and Decision-Making for Environmental Management.

References

- [Blanco *et al.*, 2009] J.L. Blanco, J. Gonzalez, and A.J. Lilienthal. An efficient approach to probabilistic gas distribution mapping. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2009. Submitted to ICRA 2009.
- [Collins and Duffy, 2002] M. Collins and N. Duffy. Convolution kernels for natural language. *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, 1:625–632, 2002.
- [Driessens *et al.*, 2006] K. Driessens, J. Ramon, and T. Gärtner. Graph kernels and Gaussian processes for relational reinforcement learning. *Machine Learning*, 2006.
- [DustBot, 2008] DustBot. DustBot - Networked and Cooperating Robots for Urban Hygiene. <http://www.dustbot.org>, 2008.
- [Frese, 2006] U. Frese. Treemap: An $o(\log n)$ algorithm for indoor simultaneous localization and mapping. *Journal of Autonomous Robots*, 21(2):103–122, 2006.
- [Grisetti *et al.*, 2007] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics*, 23(1):34–46, 2007.
- [Hayes *et al.*, 2002] A.T. Hayes, A. Martinoli, and R.M. Goodman. Distributed Odor Source Localization. *IEEE Sensors Journal, Special Issue on Electronic Nose Technologies*, 2(3):260–273, 2002.
- [Ishida *et al.*, 1998] H. Ishida, T. Nakamoto, and T. Moriizumi. Remote Sensing of Gas/Odor Source Location and Concentration Distribution Using Mobile System. *Sensors and Actuators B*, 49:52–57, 1998.
- [Kersting *et al.*, 2007] K. Kersting, C. Plagemann, P. Pfaff, and W. Burgard. Most likely heteroscedastic gaussian process regression. In *International Conference on Machine Learning (ICML)*, Corvallis, Oregon, USA, March 2007.
- [Kolmogoroff, 1941] A. Kolmogoroff. Interpolation und extrapolation von stationären zufälligen folgen. (russian. german. *Bull. Acad. Sci. URSS, Ser. Math.*, 5:3–14, 1941.
- [Le *et al.*, 2005] Q.V. Le, A.J. Smola, and S. Canu. Heteroscedastic gaussian process regression. In *Proceedings of the 22nd international conference on Machine learning*, pages 489–496, New York, NY, USA, 2005. ACM Press.
- [Lilienthal and Duckett, 2004] A. Lilienthal and T. Duckett. Building Gas Concentration Gridmaps with a Mobile Robot. *Robotics and Autonomous Systems*, 48(1):3–16, 2004.
- [Lilienthal *et al.*, 2006] A. Lilienthal, A. Loutfi, and T. Duckett. Airborne Chemical Sensing with Mobile Robots. *Sensors*, 6:1616–1678, 2006.
- [Lilienthal *et al.*, 2007] A. Lilienthal, A. Loutfi, J.L. Blanco, C. Galindo, and J. Gonzalez. A rao-blackwellisation approach to gdm-slam: Integrating slam and gas distribution mapping. In *Proc. of the European Conference on Mobile Robots (ECMR)*, pages 126–131, 2007.
- [Marques *et al.*, 2005] Lino Marques, André Martins, and A. T. de Almeida. Environmental monitoring with mobile robots. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3624–3629, 2005.
- [Meeds and Osindero, 2006] E. Meeds and S. Osindero. An alternative infinite mixture of gaussian process experts. In *Advances in Neural Information Processing Systems*, 2006.
- [Murlis *et al.*, 1992] J. Murlis, J. S. Elkington, and R. T. Carde. Odor Plumes and How Insects Use Them. *Annual Review of Entomology*, 37:505–532, 1992.
- [O’Hagan, 1978] A. O’Hagan. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society B*, 40(1), 1978.
- [Paciorek and Schervish, 2003] Christopher J. Paciorek and Mark J. Schervish. Nonstationary Covariance Functions for Gaussian Process Regression. In *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, 2003.
- [Plagemann *et al.*, 2008] C. Plagemann, K. Kersting, and W. Burgard. Nonstationary gaussian process regression using point estimates of local smoothness. In *Proc. of the European Conference on Machine Learning (ECML)*, Antwerp, Belgium, 2008.
- [Purnamadjaja and Russell, 2005] A.H. Purnamadjaja and R.A. Russell. Congregation Behaviour in a Robot Swarm Using Pheromone Communication. In *Proc. of the Australian Conf. on Robotics and Automation*, 2005.
- [Pyk *et al.*, 2006] P. Pyk, S. Bermúdez Badia, U. Bernardet, P. Knüsel, M. Carlsson, J. Gu, E. Chanie, B.S. Hansson, T.C. Pearce, and P.F. Verschure. An Artificial Moth: Chemical Source Localization Using a Robot Based Neuronal Model of Moth Optomotor Anemotactic Search. *Autonomous Robots*, 20:197–213, 2006.

-
- [Rasmussen and Ghahramani, 2002] C.E. Rasmussen and Z. Ghahramani. Infinite mixtures of gaussian process experts. In *Advances in Neural Information Processing Systems 14*, 2002.
- [Rasmussen and Williams, 2006] C. E. Rasmussen and C. K.I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
- [Rasmussen, 1996] C.E. Rasmussen. *Evaluation Of Gaussian Processes And Other Methods For Non-Linear Regression*. PhD thesis, Graduate Department of Computer Science, University of Toronto, 1996.
- [Rasmussen, 2006] C.E. Rasmussen. Minimize. <http://www.kyb.tuebingen.mpg.de/b/people/carl/code/minimize>, 2006.
- [Roberts and Webster, 2002] P.J.W. Roberts and D.R. Webster. Turbulent Diffusion. In H. Shen, A. Cheng, K.-H. Wang, M.H. Teng, and C. Liu, editors, *Environmental Fluid Mechanics - Theories and Application*. ASCE Press, Reston, Virginia, 2002.
- [Schmidt and O'Hagan, 2003] A.M. Schmidt and A. O'Hagan. Bayesian inference for nonstationary spatial covariance structure via spatial deformations. *JRSS, series B*, 65:745–758, 2003.
- [Smola and Bartlett, 2000] A.J. Smola and P.L. Bartlett. Sparse greedy gaussian process regression. In *NIPS*, pages 619–625, 2000.
- [Snelson and Ghahramani, 2006a] E. Snelson and Z. Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems 18*, pages 1259–1266, 2006.
- [Snelson and Ghahramani, 2006b] E. Snelson and Z. Ghahramani. Variable noise and dimensionality reduction for sparse gaussian process es. In *Uncertainty in Artificial Intelligence*, 2006.
- [Stachniss *et al.*, 2008] C. Stachniss, C. Plagemann, A. Lilienthal, and W. Burgard. Gas distribution modeling using sparse gaussian process mixture models. In *Proc. of Robotics: Science and Systems (RSS)*, Zurich, Switzerland, 2008. To appear.
- [Tresp, 2000] V. Tresp. Mixtures of gaussian processes. In *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, 2000.
- [Wandel *et al.*, 2003] M. Wandel, A. Lilienthal, T. Duckett, U. Weimar, and A. Zell. Gas distribution in unventilated indoor environments inspected by a mobile robot. In *Proc. of the Int. Conf. on Advanced Robotics (ICAR)*, pages 507–512, 2003.
- [Wiener, 1964] N. Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. The MIT Press, 1964.