Gas Distribution Modeling using Sparse Gaussian Process Mixture Models

Cyrill Stachniss[1] Christian Plagemann[1] Achim Lilienthal[2] Wolfram Burgard[1]

[1]*Albert-Ludwigs-University Freiburg, Dept. for Computer Science, D-79110 Freiburg, Germany*
[2]*Örebro University, AASS, Dept. of Technology, S-70182 Örebro, Sweden*
{stachnis, plagem, burgard}@informatik.uni-freiburg.de, achim.lilienthal@tech.oru.se

***Abstract*— In this paper, we consider the problem of learning a two dimensional spatial model of a gas distribution with a mobile robot. Building maps 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 present an approach that formulates this task as a regression problem. To deal with the specific properties of typical gas distributions, we propose a sparse Gaussian process mixture model. This allows us to accurately represent the smooth background signal as well as areas of high concentration. We integrate the sparsification of the training data into an EM procedure used 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. We demonstrate that our models are well suited for predicting gas concentrations at new query locations and that they outperform alternative methods used in robotics to carry out in this task.**

***Index Terms*— Gas distribution modeling, gas sensing, Gaussian processes, mixture models**

I. Introduction

Gas distribution modeling has important applications in industry, science, and every-day life. Mobile robots equipped with gas sensors are deployed, for example, for pollution monitoring in public areas [1], surveillance of industrial facilities producing harmful gases, or inspection of contaminated areas within rescue missions.

Although humans have a natural odor sensor, it is hard for us to build a spatial representation of a sensed gas distribution. Building gas distribution maps is actually a challenging task due to the chaotic nature of gas dispersal. 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 to achieve 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 [16]. Especially this chaotic system of localized patches of gas makes the modeling problem a hard one. In addition to that, gas sensors provide information about a small spatial region only since gas sensor measurements require direct interaction between the sensor surface and the analyze molecules. This makes gas sensing different to perceiving the environment with laser range finders or other popular robotic sensors.

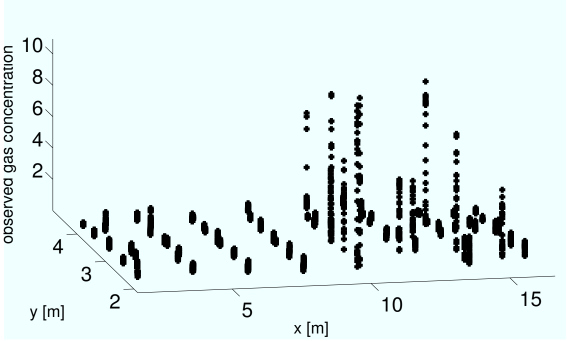


Fig. 1. Gas concentration measurements acquired by a mobile robot in a corridor. The distribution consists of a rather smooth "background" signal and several peaks indicating high gas concentrations.

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 indicating 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. Since performing measurements is a comparably costly operation, one is also interested in reducing the number of samples needed to build a representation. It is important to note that the noise is dominated by the large fluctuations of the instantaneous gas distribution and not by the electronic noise of the gas sensors. 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. Thus, the data likelihood in combination with cross validation is the standard criterion to evaluate such a model.

Simple spatial averaging, which represents a straightforward approach to the modeling problem, disregards the different nature of the background noise 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 tractability, we treat gas distribution mapping as a two-dimensional regression problem. We derive a solution by means of a sparse mixture model of Gaussian process experts [21] 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 at locations as noisy samples from a time-constant distribution. 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 [22].

While existing approaches to gas distribution mapping such as local averaging [6, 11], kernel extrapolation techniques [7], or standard GP models represent the average concentration per location only, our mixture model explicitly distinguishes different components of the distribution, i.e., concentration layers varying smoothly due to dispersion processes versus those containing localized *patches* of gas. This leads to a more accurate prediction of the gas concentration. Our model actually allows us to do both, computing the average gas concentration per location (as existing models supply) as well as the multi-modal predictive densities.

The contribution of this paper is a novel approach that learns gas distribution models from sensor data using a sparse Gaussian process mixture model. As a by-product, we present an algorithm that learns a GP mixture model and simultaneously reduces the model complexity in order to achieve an efficient representation even for large data sets. Our technique provides gas concentration estimates for each location in space and also the corresponding predictive uncertainties. The mixture model allows us to improve the gas concentration estimate close to the boundaries and in areas with high gas concentration compared to standard models. As we will demonstrate in experiments carried out with a real robot, our model has a lower mean squared error and a higher data likelihood than other methods and thus allows to more accurately predict gas concentration at query locations.

This paper is organized as follows. After a discussion of related work, we introduce in Sec. III Gaussian processes for regression. Then, Sec. IV explains our approach to learn a sparse GP mixture to model gas distributions from observations. Finally, we present the experimental evaluation of our work with a real mobile robot.

II. Related Work

A straightforward method to create a representation of the time-averaged concentration field is to perform measurements over a prolonged time with a grid of gas sensors. Equidistant gas sensor locations can be used to represent the average concentration values directly on a grid map. This method, though with partially simultaneous measurements, was applied by Ishida *et al.* [6]. A similar method was used in [11] 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.* [12] 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., uniform delivery and removal of the analyze gas and stable environmental conditions. Thus, the experiments of Pyk *et al.* were performed in a wind tunnel with a constant airflow and a uniform gas source. To make predictions at locations different from the measurement points, they apply bi-cubic interpolation in the case of equidistant measurements and triangle-based cubic filtering in the case where the measurement points are not equally distributed [12]. A problem with these interpolation methods is that there is no means of "averaging out" instantaneous response fluctuations at measurement locations. Even if response values were measured very close to each other, they will appear independently in the gas distribution map with interpolated values in-between. Consequently, interpolation maps tend to get more and more jagged while new measurements are added [8].

Histogram methods take the spatial correlation of concentration measurements into account because of the implicit extrapolation on the measurements by the quantization into histogram bins. Hayes *et al.* [5] suggest a two-dimensional histogram where the bins contain the accumulated 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 problem with using only binary information from the gas sensors is that much useful information about fine gradations in the average concentration is discarded. A further disadvantage of histogram methods for gas distribution modeling is their dependency on the bin size and that they require perfectly even coverage of the inspected area.

Kernel extrapolation gas distribution mapping, which can be seen as an extension of histogram methods, was introduced by Lilienthal and Duckett [7]. 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 [8], this method has also an analogy with non-parametric estimation of density functions using a Parzen window method.

Model-based approaches as in Ishida *et al.* [6] infer the parameters of an analytical gas distribution model from the measurements. They depend crucially on the underlying model. Complex numerical models based on fluid dynamics simulations are computationally expensive and depend sensitively on accurate knowledge of the state of the environment (boundary conditions) which is not available in practical situations. Simpler analytical models, on the other hand, often rest on rather unrealistic assumptions and are of course only applicable for situations in which the model assumptions hold. Model-based approaches also rely on well-calibrated gas sensors and an established understanding of the sensor-environment interaction.

The majority of approaches proposed in the literature create a two-dimensional representation and represent time-constant structures in the gas distribution. Also the effort (either in terms of time consumption or the number of sensors) of the model-free approaches to converge to a stable representation, scales quadratically with the size of the environment. None of the approaches suggested so far models the variance together with the time-average of the concentration field.

In contrast to those approaches, we apply Gaussian processes in a mixture model setting to learn probabilistic gas distribution maps. GPs allow us to model the dependency between nearby locations 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 a predictive variance. Our mixture model can furthermore model sharp boundaries of areas with high gas concentration.

Gaussian processes (GPs) are a non-parametric method frequently used to solve regression and classification problems [13]. A drawback of the standard GP approach is its computational complexity. However, several methods for learning sparse GP models [18, 19] have been presented that overcome this limitation and lead to a near-linear complexity [19]. Tresp [21] introduced a mixture model of GP experts to better deal with spatially varying properties in the data. Extensions of this technique using infinite mixtures have been proposed by Rasmussen and Ghahramani [15] and Meeds and Osindero [9].

GPs have already received considerable attention within the robotics community. Schwaighofer *et al.* [17] introduced a positioning system for cellular networks based on Gaussian processes. Brooks *et al.* [2] proposed a Gaussian process model in the context of appearance-based localization with an omni-directional camera. Ferris *et al.* [3] applied Gaussian processes to locate a mobile robot from wireless signal strength. Related Bayesian regression approaches have been also followed for example by Ting *et al.* [20] to identify rigid body dynamics and Grimes *et al.* [4] to learn imitative whole-body motions.

III. Gaussian Processes for Regression

The general gas distribution mapping problem, given a set of gas concentration measurements $y_{1:n}$ acquired at locations $\mathbf{x}_{1:n}$, is to learn a predictive model $p(y_* \mid \mathbf{x}_*, \mathbf{x}_{1:n}, y_{1:n})$ for gas concentrations y_* at a query location $\mathbf{x}_*$. We address this estimation problem as a regression problem. Gaussian processes (GPs) offer a flexible way of solving such regression problems [13]. GPs are a "non-parametric" method, since no parametric form of the underlying function $\mathbf{x} \mapsto y$ is assumed. The model is represented directly using the given training data. GPs can be seen as a generalization of the Gaussian probability distribution to a distribution over functions. A GP for real-valued functions f is defined by a mean function $m(\cdot)$ and a covariance function $k(\cdot,\cdot)$

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})] & (1)\\ k(\mathbf{x}_p, \mathbf{x}_q) &= \mathbb{E}[(f(\mathbf{x}_p) - m(\mathbf{x}_p))(f(\mathbf{x}_q) - m(\mathbf{x}_q))]. & (2)\end{aligned}$$

In the following, we set $m(\mathbf{x}) = 0$ for simplicity of notation and apply the squared exponential covariance function

$$k(\mathbf{x}_p, \mathbf{x}_q) = \sigma_f^2 \cdot \exp\left(-\frac{1}{2}\frac{|\mathbf{x}_p - \mathbf{x}_q|^2}{l^2}\right). \quad (3)$$

Observations y obtained from the process are assumed to be affected by Gaussian noise, $y \sim \mathcal{N}(m(\mathbf{x}), \sigma_n^2)$. The variables $\Phi = \{\sigma_f, l, \sigma_n\}$ are the so-called hyperparameters of the process which have to be learned from data.

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 $X = [\mathbf{x}_1; \ldots; \mathbf{x}_n]$ be the $n \times d$ matrix of the inputs and X_* be defined analogously for multiple test data points. In the GP model, any finite set of samples is jointly Gaussian distributed

$$\begin{bmatrix} \mathbf{y} \\ f(X_*) \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right), \quad (4)$$

where $K(\cdot,\cdot)$ refers to the matrix with the entries given by the covariance function $k(\cdot,\cdot)$ and $\mathbf{y}$ the vector of the (observed) targets y_i. To actually make predictions at X_*, we obtain for the predictive mean

$$\bar{f}(X_*) := \mathbb{E}[f(X_*)] = K(X_*, X)[K(X,X) + \sigma_n^2 I]^{-1}\mathbf{y} \quad (5)$$

and for the (noise-free) predictive variance

$$\begin{aligned} \mathbb{V}[f(X_*)] = {} & K(X_*, X_*) \\ & -K(X_*, X)[K(X,X) + \sigma_n^2 I]^{-1} K(X, X_*), \quad (6)\end{aligned}$$

where 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 $\mathbb{V}[f(X_*)]$.

GPs are a sound mathematical framework with many practical applications. The standard GP model as described above, however, has two major limitations in our problem domain. First, the computational complexity is high, since to compute the predictive variance given in Eq. (6), one needs to invert the matrix $K(X,X) + \sigma_n^2 I$, which 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. IV, be achieved by artificially limiting the training data set in a way that introduces small loss in the data likelihood of the whole training set 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 high-concentration "packets" of gas. In the following, we address this issue by deriving a *mixture model* of Gaussian processes.

A. Mixtures of Gaussian Process Models

The GP mixture model [21] 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 X_*). Let $\{\mathcal{GP}_1, \ldots, \mathcal{GP}_m\}$ 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 the $\mathcal{GP}_i$ at $\mathbf{x}_*$. The likelihood of observing y_* in such as model is thus given by

$$h(\mathbf{x}_*) := p(y_* \mid \mathbf{x}_*) = \sum_{i=1}^{m} P(z(\mathbf{x}_*) = i) \cdot \mathcal{N}_i(y_*; \mathbf{x}_*), \quad (7)$$

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

$$\bar{h}(\mathbf{x}_*) := \mathbb{E}[h(\mathbf{x}_*)] = \sum_{i=1}^{m} P(z(\mathbf{x}_*) = i) \cdot \bar{f}_i(\mathbf{x}_*) \tag{8}$$

and the corresponding variance is computed as

$$\begin{aligned} \mathbb{V}[h(\mathbf{x}_*)] \;\; = \;\; & \sum_{i=1}^{m} (\mathbb{V}[f_i(\mathbf{x}_*)] + (\bar{f}_i(\mathbf{x}_*) - \bar{h}(\mathbf{x}_*))^2) \\ & \cdot P(z(\mathbf{x}_*) = i). \end{aligned} \tag{9}$$

IV. Learning the Mixture 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 [21] describes an approach based on Expectation Maximization (EM) for solving this task. We take his approach, but also seek to minimize the model complexity 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 $n = 1000$ data points and the standard GP model (see Sec. III) 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 the usage of the so called *sparse GPs*. Sparse GPs [18, 19] use a reduced set of inputs to approximate the full space. This new set can be either a subset of the original inputs [18] or a set of new *pseudo-inputs* [19] which are obtained using an optimization procedure. This reduces the complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(nm^2)$ with $m \ll n$, which in practice results in a nearly linear complexity. In this section, we describe a greedy forward-selection algorithm integrated into the EM-learning procedure which achieves a sparse mixture model while also maximizing the data likelihood of the whole training set $\mathcal{D}$.

A. Initializing the Mixture Components

In a first step, we subsample n_1 data points and learn a standard GP for this set. This model $\mathcal{GP}_1$ constitutes the first mixture component. To cover areas of gas concentrations that are poorly modeled by this initial model, we learn an "error GP" which models the absolute differences between a set of target values and the predictions of $\mathcal{GP}_1$. We then sample points according to the error GP and use them as the initialization for the next mixture component. In this way, the new mixture is initialized with the data points that are poorly approximated by the first one. This process is continued until the desired number m of model components is reached. For typical gas modeling scenarios, we found that two mixture components are often sufficient to achieve good results. In our experiments, the converged mixture models nicely reflect the bi-modal nature of gas distributions, having one smooth "background" component and a layer of locally concentrated structures as outlined in the introduction of this paper.

B. 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 the data point j corresponds to the model i. This is done by computing the marginal likelihood of each data point for all models individually. Thus, the new $P(z(\mathbf{x}_j) = i)$ is computed given the previous one as

$$P(z(\mathbf{x}_j) = i) \;\; \leftarrow \;\; \frac{P(z(\mathbf{x}_j) = i) \cdot \mathcal{N}_i(y_j; \mathbf{x}_j)}{\sum_{k=1}^{m} P(z(\mathbf{x}_j) = k) \cdot \mathcal{N}_k(y_j; \mathbf{x}_j)} \tag{10}$$

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 [21]). This is achieved by modifying Eq. (5) to

$$\bar{f}_i(X_*) \;\; = \;\; K(X_*, X)[K(X, X) + \Psi^i]^{-1}\mathbf{y}, \tag{11}$$

where Ψ^i is a matrix with

$$\Psi^i_{jj} \;\; = \;\; \frac{\sigma_n^2}{P(z(\mathbf{x}_j) = i)} \tag{12}$$

and zeros in the off-diagonal elements. Eq. (6) is updated respectively. The matrix Ψ^i allows us to consider the probabilities that the individual inputs belong to the corresponding components. The contribution of an unlikely data point to a model is reduced by increasing the data point specific noise term. If the probability, however, is one, only σ_n^2 remains as in the standard GP model.

Learning a GP model also involves the estimation of its hyperparameters $\Phi = \{\sigma_f, l, \sigma_n\}$. To estimate them for $\mathcal{GP}_i$, we first apply a variant of the hyperparameter heuristic used by Snelson and Ghahramani [19] in their open-source implementation. We extended it to incorporate the correspondence probability $P(z(\mathbf{x}_k) = i)$ into this initial guess

$$l \;\; \leftarrow \;\; \max_{\mathbf{x}_j} \; P(z(\mathbf{x}_j) = i) \; ||\mathbf{x}_j - \bar{\mathbf{x}}|| \tag{13}$$

$$\sigma_f^2 \;\; \leftarrow \;\; \frac{\sum_{j=1}^{n} P(z(\mathbf{x}_j) = i) \; (y_j - \mathbb{E}[y])^2}{\sum_{j=1}^{n} P(z(\mathbf{x}_j) = i)} \tag{14}$$

$$\sigma_n^2 \;\; \leftarrow \;\; 0.25 \cdot \sigma_f^2 \; , \tag{15}$$

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 further given this initial estimate, one could apply, for example, Rasmussen's conjugate-gradient–based optimization technique [14] to minimize the negative log marginal likelihood. In our experiments, however, this approach lead to serious overfitting and we therefore resorted to cross validation-based optimization. Concretely, we randomly sample the hyperparameters and evaluate the model accuracy according to Sec. IV-B on a separate validation set. As a sampling strategy, we draw in each even iteration new parameters 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 decreases with the iteration. In our experiments, this strategy found appropriate hyperparameters quickly while significantly reducing the risk of overfitting.

C. Learning the Gating Function

In our mixture model, the gating function defines for each data point the likelihood that it belongs to the individual mixture components. The EM algorithm learns these assignment probabilities for all inputs $\mathbf{x}_j$, maximizing the overall data likelihood. These learned hidden variables are then used to estimate the assignment at an unknown location $\mathbf{x}_*$ by means of regression. 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

$$P(z(\mathbf{x}_*) = i) \quad = \quad \frac{\exp(\bar{f}_i^z(\mathbf{x}_*))}{\sum_{j=1}^{m} \exp(\bar{f}_j^z(\mathbf{x}_*))}. \qquad (16)$$

D. Illustrating Example

We have specified all quantities that are needed to model gas distributions with sparse Gaussian process mixture models. To summarize the approach, we use a a simple, simulated, one-dimensional example.

The first part of the data points where uniformly distributed around a y value of 2 while the second part was generated with higher noise at two distinct locations. The left image of Fig. 2 depicts the standard GP learned from the input data and the right one the resulting error GP. Based on the error GP, a second mixture component is initialized and used as the input to the EM algorithm.

The individual images in Fig. 3 illustrate the iterations of the EM algorithm. They depict the two components of the mixture model. After convergence, the gating function is learned using the hidden variables reported by the EM algorithm. The learned gating function is depicted in the left image of Fig. 4 and the final GP mixture model is shown in the right image. It is obvious that this model is a better representation of the distribution than the standard GP model shown in the left image of Fig. 2 (averaged negative log likelihood of -1.70 vs. -0.24).

V. Experiments

We carried out pollution monitoring experiments in which the robot followed a predefined sweeping trajectory covering the area of interest. Along its path, the robot was stopped at a pre-defined set of grid points to carry out measurements on the spot between 10 s (outdoors) and 30 s (indoors). The spacing between the grid points was set to values between 0.5 m to 2.0 m depending on the topology of the available space. The sweeping motion was performed twice in opposite directions and the robot was driven at a maximum speed of 5 cm/s in between the stops (to reduce the risk of turbulent air flow due to the motion of the robot). The gas source was a small cup filled with ethanol.

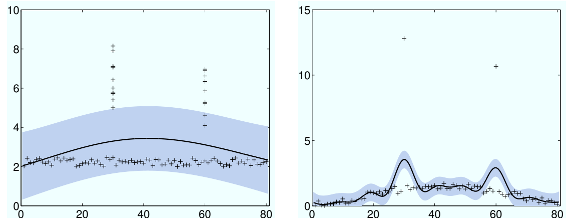


Fig. 2. Left: The standard GP used to initialize the first mixture component. Right: The error GP used to initialize the next mixture component.

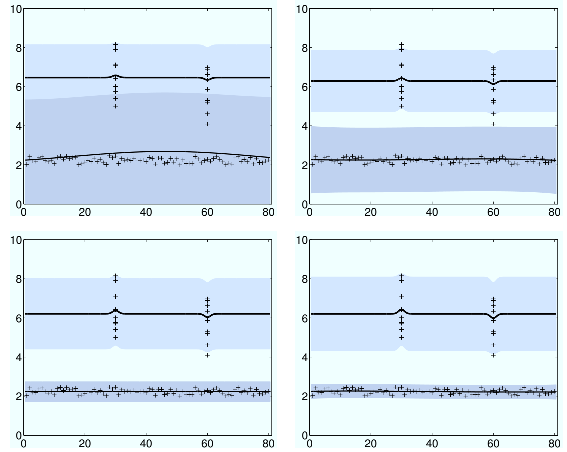


Fig. 3. Components during different iterations of the EM algorithm.

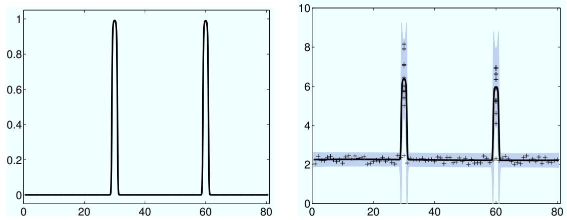


Fig. 4. Left: The learned gating function. Right: Resulting distribution of the GP mixture model.

Apart from a SICK laser range scanner used for pose correction, the robot was equipped with an electronic nose and an anemometer. The electronic nose comprises six Figaro gas sensors (2 × TGS 2600, TGS 2602, TGS 2611, TGS 2620, TGS 4161) enclosed in an aluminum tube. This tube is horizontally mounted at the front side of the robot (see also Fig. 5). The electronic nose is actively ventilated through a fan that creates a constant airflow towards the gas sensors. 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.

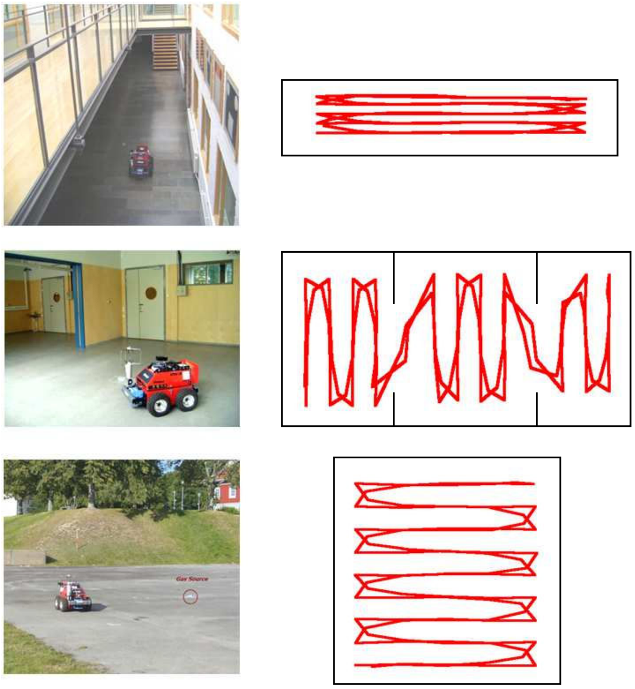


Fig. 5. Pictures of the robot inspecting three different environments as well as the corresponding sweeping trajectories.

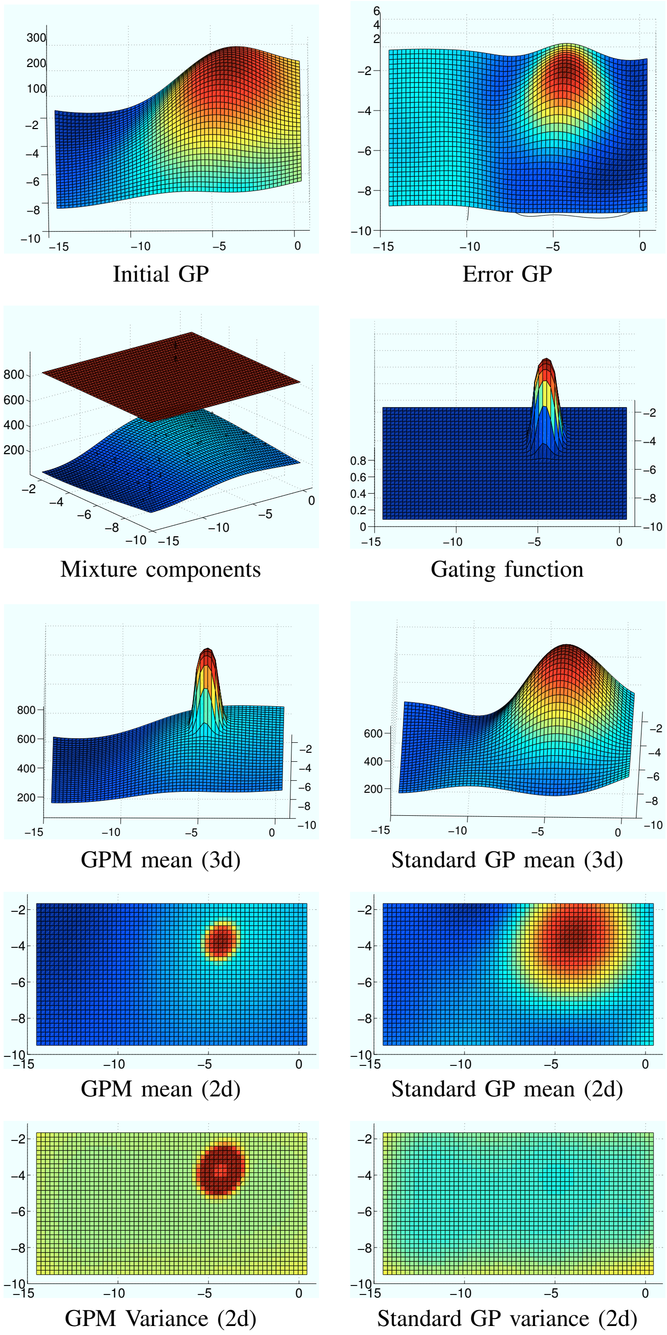
TABLE I

AVERAGED NEGATIVE LOG LIKELIHOODS OF TEST DATA POINTS GIVEN THE DIFFERENT MODELS

Dataset	GP	GPM	GPM avg
3-rooms	-1.22	-1.54	-1.50
corridor	-0.98	-1.60	-1.58
outdoor	-1.01	-1.77	-1.69

Three environments with different properties have been selected for the pollution monitoring experiments. The first experiment (*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 path of the robot is approximately 14×6 m^2. There is very 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 14×2 m^2. 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 8×8 m^2 region that is part of a much bigger open area.

We used the raw sensor readings in all three environments and applied our approach to learn gas distribution models. In the experiments shown here, the robot moved through the environment twice. Therefore, we used the first run for learning the model and the second one for evaluating it. For a comparison with our technique, we also computed a gas distribution model using a standard GP. We furthermore compared our mean estimates to the one of the grid-based method with interpolation and the kernel extrapolation technique.



Initial GP

Error GP

Mixture components

Gating function

GPM mean (3d)

Standard GP mean (3d)

GPM mean (2d)

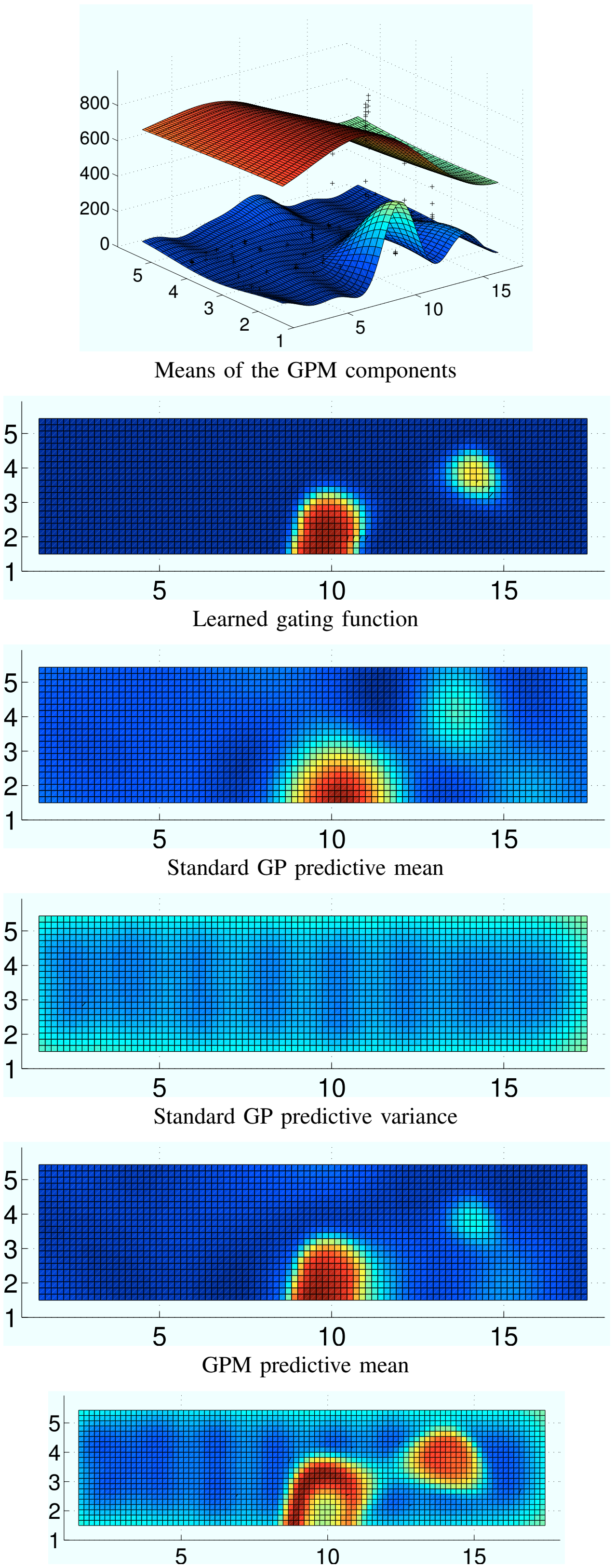
Standard GP mean (2d)

GPM Variance (2d)

Standard GP variance (2d)

Fig. 6. 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 reflect medium, and red refers to high values.

Fig. 6 depicts 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 sub-sampled training set which serves as the first mixture component. The right image 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. After convergence, the gating function is learned based on the hidden variables reported by the EM (right image, second row). The left image in the third row shows the final mean prediction of our mixture model. As can be seen, the "background" distribution is smoothly modeled while at the



Means of the GPM components

Learned gating function

Standard GP predictive mean

Standard GP predictive variance

GPM predictive mean

GPM predictive variance

Fig. 7. Models learned from concentration data recorded in the corridor environment (see Fig. 1 for the raw data). The gas source was placed at the location 10, 3. The standard GP and our GPM model provide similar mean estimates. Our approach, however, provides a better predictive uncertainty and thus a higher likelihood given the test data (see Tab. I).

same time the gas concentration peak close to the gas source has a sharp boundary. In contrast to this, the standard GP learned using the same data is unable to provide an appropriate estimate since the area around the peak is to smoothed too much.

Tab. I summarizes the negative log likelihoods of the test data (second part of the dataset) given our mixture model as well as the standard GP model. We provide two likelihoods for our model, the one given in Eq. (7) (called 'GPM' in the table) and the one computed based on the averaged prediction specified in Eq. (8) and Eq. (9) (called 'GPM avg'). As can be seen, our GPM method outperforms the standard GP model in all our experiments since it provides the best data likelihood. Note that we repeated the experiment 10 times and the t-test shows that the results are significant.

By considering the 2d plots in the last two rows of Fig. 6, two reasons for this fact can be observed easily. First, as already mentioned before, the standard GP smoothes too much in the area close to the gas source while this smoothing is fine for the rest of the scene. Second, the variance around the source is too small (standard GPs assumes constant noise for all inputs).

In the corridor experiment, the area of high gas concentration was mapped appropriately also by the standard GP, but again the variance was too small close to the area of high gas concentration. This can be observed by considering Fig. 7. In contrast to this, our GPM model provides a high variance in this area – which actually models the observations in a more precise way. Similar results are obtained in the outdoor dataset. Mean and variance predictions of the standard GP and our model are provided in Fig. 9.

In all our experiments, we limited the number of data points in the reduced input set to $n_1 = 100$ (taken from the first part of the datasets). 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 and as a result the overall computation time to learn our model including cross validation using unoptimized Matlab code on a notebook computer takes around 1 minute for all datasets shown above.

Finally, we compared the mean estimates of our mixture model to the results obtained with the method of Lilienthal and Duckett [7] as well as with the standard approach of using a grid in combination with interpolation. The results of this comparison is shown in Fig. 8. As can be seen, our method outperforms both alternative methods.

VI. Conclusions

In this paper, 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 model complexity by reducing the training set in order to achieve an efficient representation even for a large number of observations. This overcomes the major drawback of GPs, their high

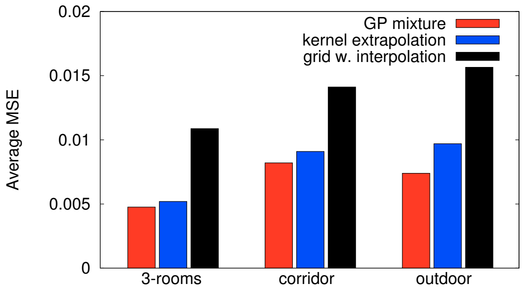


Fig. 8. Mean squared error of the GP mixture model mean and the kernel extrapolation technique and the grid approximation with interpolation.

computational complexity. 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 [7] provide similar mean gas concentration estimates, their approach as well as the majority of techniques in the field, however, lack the ability of estimating their predictive uncertainties.

Despite this encouraging results, there is space for further optimizations. Considering non-stationary kernels [10] might further improve the estimates or might serve as an alternative to explictly modeling mixtures. In addition, we are currently exploring the possibility to model the diffusion in high concentration areas by smoothing the gating function over time.

Acknowledgment

This work has partly been supported by the DFG under contract number SFB/TR-8 as well as by the EC under contract number FP6-IST-34120-muFly and FP6-2005-IST-6-RAWSEEDS.

References

[1] DustBot - Networked and Cooperating Robots for Urban Hygiene. http://www.dustbot.org.

[2] A. Brooks, A. Makarenko, and B. Upcroft. Gaussian process models for sensor-centric robot localisation. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2006.

[3] B. Ferris, D. Haehnel, and D. Fox. Gaussian processes for signal strength-based location estimation. In *Proceedings of Robotics: Science and Systems*, 2006.

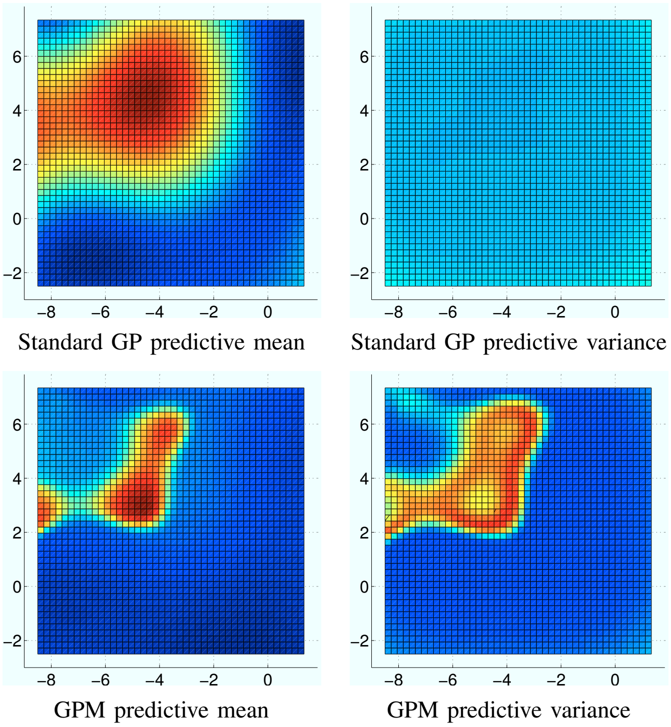
[4] D. Grimes, R. Chalodhorn, and R. Rao. Dynamic imitation in a humanoid robot through nonparametric probabilistic inference. In *Proceedings of Robotics: Science and Systems*, 2006.

[5] 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.

[6] 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.

[7] A. Lilienthal and T. Duckett. Building Gas Concentration Gridmaps with a Mobile Robot. *Robotics and Autonomous Systems*, 48(1):3–16, 2004.

[8] A. Lilienthal, A. Loutfi, and T. Duckett. Airborne Chemical Sensing with Mobile Robots. *Sensors*, 6:1616–1678, 2006.



Standard GP predictive mean

Standard GP predictive variance

GPM predictive mean

GPM predictive variance

Fig. 9. Outdoor dataset of a 8 m by 8 m area with an ethanol source in the center and airflow. approximatively from south-east to north-west.

[9] E. Meeds and S. Osindero. An alternative infinite mixture of gaussian process experts. In *Advances in Neural Information Processing Systems*, 2006.

[10] 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.

[11] 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.

[12] P. Pyk et al. An Artificial Moth: Chemical Source Localization Using a Robot Based Neuronal Model of Moth Optomotor Anemotactic Search. *Autonomous Robots*, 20:197–213, 2006.

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

[14] C.E. Rasmussen. Minimize. http://www.kyb.tuebingen.mpg.de/bs/people/carl/code/minimize, 2006.

[15] C.E. Rasmussen and Z. Ghahramani. Infinite mixtures of gaussian process experts. In *Advances in Neural Information Processing Systems 14*, 2002.

[16] 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.

[17] A. Schwaighofer, M. Grigoras, V. Tresp, and C. Hoffmann. Gpps: A gaussian process positioning system for cellular networks. In *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, 2003.

[18] A.J. Smola and P.L. Bartlett. Sparse greedy gaussian process regression. In *NIPS*, pages 619–625, 2000.

[19] E. Snelson and Z. Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems 18*, pages 1259–1266, 2006.

[20] J. Ting, M. Mistry, J. Peters, S. Schaal, and J. Nakanishi. A bayesian approach to nonlinear parameter identification for rigid body dynamics. In *Proceedings of Robotics: Science and Systems*, 2006.

[21] V. Tresp. Mixtures of gaussian processes. In *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, 2000.

[22] 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.