

Gaussian Beam Processes: A Nonparametric Bayesian Measurement Model for Range Finders

Christian Plagemann Kristian Kersting Patrick Pfaff Wolfram Burgard

Albert-Ludwigs-University of Freiburg, Department for Computer Science, 79110 Freiburg, Germany
 {plagem,kersting,pfaff,burgard}@informatik.uni-freiburg.de

Abstract—In probabilistic mobile robotics, the development of measurement models plays a crucial role as it directly influences the efficiency and the robustness of the robot’s performance in a great variety of tasks including localization, tracking, and map building. In this paper, we present a novel probabilistic measurement model for range finders, called Gaussian beam processes, which treats the measurement modeling task as a nonparametric Bayesian regression problem and solves it using Gaussian processes. The major benefit of our approach is its ability to generalize over entire range scans directly. This way, we can learn the distributions of range measurements for whole regions of the robot’s configuration space from only few recorded or simulated range scans. Especially in approximative approaches to state estimation like particle filtering or histogram filtering, this leads to a better approximation of the true likelihood function. Experiments on real world and synthetic data show that Gaussian beam processes combine the advantages of two popular measurement models.

I. INTRODUCTION

Acquiring, interpreting, and manipulating information from sensors is one of the fundamental tasks within mobile robotics. For instance, based on models for the robot’s kinematics and perception, a robot might be asked to perform tasks such as building a map of the environment, determining its precise location within a map, or navigating to a particular place. In designing robots to operate in the real world, one cannot avoid dealing with the issue of uncertainty. Uncertainty arises from sensor limitations, noise, and the fact that most complex environments can only be represented and perceived in a limited way. It is therefore not surprising that state-of-the-art approaches build on probabilistic foundations and model the robot’s perception as a probability density $p(\mathbf{z}|\mathbf{x})$, where \mathbf{z} is an observation and \mathbf{x} denotes the state of the robot, e.g., its position relative to the environment.

Among the most widely used types of sensors are range sensors such as laser range finders and sonar sensors. The popularity of range sensors in research and industry is due to the fact that spatial information about the environment can directly be acquired and that state-of-the-art sensor hardware is quite accurate and reliable. Range sensors measure distances r_i to nearby objects along certain directions α_i (possibly multivariate bearing angles) relative to the sensor. Hence, for a vector $\mathbf{r} = (r_1, \dots, r_m)$ of distance measurements with corresponding bearing angles $\mathcal{A} = (\alpha_1, \dots, \alpha_m)$, the likelihood function $p(\mathbf{z}|\mathbf{x})$ can be rewritten as $p(\mathbf{r}|\mathcal{A}, \mathbf{x})$.

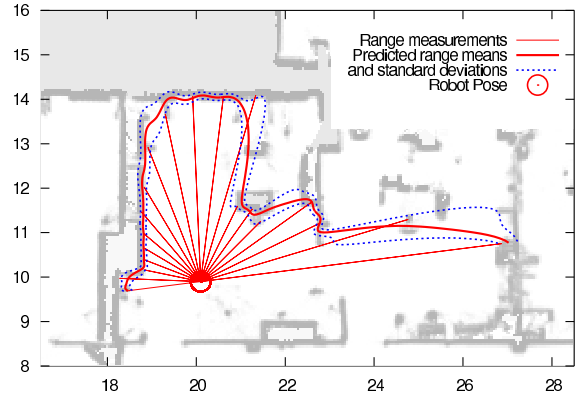


Fig. 1. The predictive distribution of range measurements for an uncertain robot pose within an office environment (heading angle varies by $\pm 5^\circ$). Scales are given in meters. The straight, red lines depict one possible range scan in this setting.

In this paper, we propose a novel, generative model for $p(\mathbf{r}|\mathcal{A}, \mathbf{x})$, called Gaussian beam processes (GBP), which takes a nonparametric Bayesian regression view on the measurement modeling task. We treat the measurements $\mathbf{z} = (r_i, \alpha_i)_{i=1}^m$ as a set of samples from a stochastic process $p(r|\alpha)$ and assume the process to be a Gaussian process [1], i.e., any finite collection of random variables r_j is assumed to have a joint Gaussian distribution. Learning in this framework means recording or simulating a training set of range scans and adjusting a predefined covariance function accordingly. This can be done online while the robot is operating or offline.

We put special emphasis on the application of GBPs to mobile robot localization, but nevertheless present the model in a general form that should be useful in many other applications for range sensor. A major benefit of our model in localization tasks is that it naturally also allows to estimate the distribution $p(\mathbf{r}|\mathcal{A}, \mathcal{X})$ of range measurements for a region \mathcal{X} of the pose space. As an example, consider Figure 1. It shows the predictive distribution of range measurements for a mobile robot with an uncertain heading angle, i.e., its location is fixed, the orientation angle, however, is known only up to $\pm 5^\circ$. It can be clearly seen from the visualized standard deviations of the range predictions that the model accurately identifies the distances that can be predicted with high confidence despite of the angular uncertainty in the sensor pose. The ability to

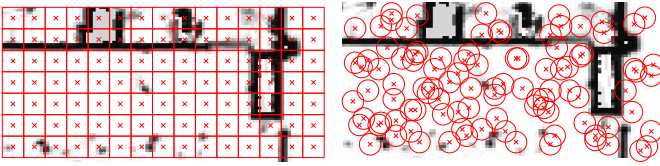


Fig. 2. The posterior distribution of robot poses is typically approximated by discretizing the pose space (left) or by sampling it (right).

learn and represent such distributions is of special value in applications in which the posterior is approximated using a discrete set of pose hypotheses. In histogram filtering, for example, the pose space is partitioned into a finite set of grid cells (see the left diagram of Figure 2). With the GBP model, we can estimate the observation likelihood $p(\mathbf{z}|\mathcal{X})$ for a whole grid cell \mathcal{X} directly rather than having to numerically approximate $\frac{1}{|\mathcal{X}|} \int_{\mathcal{X}} p(\mathbf{z}|\mathbf{x}) d\mathbf{x}$ from point estimates $p(\mathbf{z}|\mathbf{x})$, $\mathbf{x} \in \mathcal{X}$, of the likelihood function. This ability is also useful for particle filtering in which the posterior is represented by a finite set of weighted samples. It is a well-known fact that highly peaked likelihood functions have to be regularized in practical applications, because the number of particles is limited. A popular way of doing this is to locally average the likelihood function in the vicinity of each particle (see the right diagram of Figure 2), rather than taking point estimates at the exact particle locations only. Additionally, the Gaussian process treatment offers the following distinct benefits:

- The model is fully predictive as it is able to predict ranges at arbitrary bearing angles, i.e., also for angles in between two beams of an actual scan and for beams that have been classified as erroneous. For such predictions, the model also yields the predictive uncertainties.
- Neither the number of range measurements per scan nor their bearing angles have to be fixed beforehand.
- By representing correlations between adjacent beams using parameterized covariance functions, only few recorded or simulated range scans $(\mathcal{A}_j, \mathbf{r}_j)$ are required to learn an accurate model.
- Gaussian processes are mathematically well-established. There exists a great pool of methods for learning, likelihood evaluation, and prediction.

The paper is organized as follows. In Section II, we discuss related work. Section III presents our Gaussian beam process model. In Section IV, we describe its application to Monte Carlo localization and present the results of extensive evaluations on real world and synthetic data.

II. RELATED WORK

Probabilistic measurement models (or observation models) are conditional probability distributions $p(\mathbf{z}|\mathbf{x})$ that characterize the distribution of possible sensor measurements \mathbf{z} given the state \mathbf{x} of the system. In the context of mobile robot localization with laser range finders, for instance, \mathbf{x} denotes the three-dimensional pose (2D location and orientation) of the robot and \mathbf{z} stands for a vector of range readings received from

the sensor hardware. Each component (α_i, r_i) of \mathbf{z} contains a distance measurement r_i along a beam with an angle α_i relative to the sensor. The angular density of measurements and the measurement noise can vary greatly depending on the type of sensor used. The task of sensor modeling is challenging as one has to take various types of uncertainty into account such as sensor noise (e.g., due to atmospheric effects), map and pose uncertainty (e.g., caused by discretization errors), and environmental dynamics like object displacements.

Feature-based approaches typically extract a set of features from the range scan \mathbf{z} and match them to features contained in an environmental model in order to obtain $p(\mathbf{z}|\mathbf{x})$. Whereas such approaches have been proven to be robust in various applications, they assume that the features are known beforehand and that they can be extracted reliably, which might be hard in unstructured or cluttered environments. Alternative approaches directly operate on the dense measurements and therefore are applicable even in situations in which the relevant features are unknown.

Beam-based models consider each value r_i of the measurement vector \mathbf{z} as a separate range measurement and represent its one-dimensional distribution by a parametric function depending on the expected range measurement in the respective beam direction (see Fox *et al.* [2] for example). Such models are closely linked to the geometry and the physics involved in the measurement process. In the remainder of this paper, we will also denote such models as ray cast models because they rely on ray casting operations within an environmental model, e.g., an occupancy grid map, to calculate the expected beam lengths. As a major drawback, the traditional approach assumes independent beams, which leads to overly peaked likelihood functions when one increases the number of beams per measurement (e.g., to increase the spatial resolution). In practice, this problem is dealt with by sub-sampling of measurements, by introducing minimal likelihoods for beams, by inflating the measurement uncertainty [3], or by other means of regularization of the resulting likelihoods (see, e.g., Arulampalam *et al.* [4]).

Correlation-based methods typically build local maps from consecutive scans and correlate them with a global map [5, 6]. A simple and effective approach that is also associated to this class of models is the so-called likelihood fields model or end point model [7]. Here, the likelihood of a single range measurement is a function of the distance of the respective end point of a beam to the closest obstacle in the environment. Like in the ray cast model, each beam is treated independently. This model lacks a physical explanation, as it can basically “see through walls”, but it is more efficient than ray cast models and works well in practice.

Work that specifically dealt with peaked measurement models include Pfaff *et al.* [8], who adapt the smoothness of the likelihood model depending on the region covered by the individual particles, Fox *et al.* [9], and Kwok *et al.* [10], who adapt the number of particles depending on the progress of the localization process and computational power. These approaches have been developed independently from specific

measurement models and should be directly applicable to GBPs as well. Finally, GBPs are related to Gutierrez-Osuna *et al.*'s [11] neural networks approach to modeling the measurement of an ultrasonic range sensor, which in contrast to GBPs assumes scans of fixed size.

Our GBP model, which is detailed in the following section, seeks to combine the advantages of the above mentioned approaches. It represents correlations between adjacent beams using covariance functions, it respects the geometry and physics of the measurement process, and it provides a natural and intuitive way of regularizing the likelihood function.

Gaussian processes have already received considerable attention within the robotics community. Schwaighofer *et al.* [12] introduced a positioning system for cellular networks based on Gaussian processes. Brooks *et al.* [13] proposed a Gaussian process model in the context of appearance-based localization with an omni-directional camera. Ferris *et al.* [14] applied Gaussian processes to locate a mobile robot from wireless signal strength. Plagemann *et al.* [15] used Gaussian processes to detect failures on a mobile robot. Indeed, Bayesian (regression) approaches have been also followed for example by Ting *et al.* [16] to identify rigid body dynamics and Grimes *et al.* [17] to learn imitative whole-body motions.

GBPs are also related to the theory and application of Gaussian processes. Most Gaussian processes methods rely on the assumption that the noise level is uniform throughout the domain [18], or at least, its functional dependency is known beforehand [19]. Gaussian beam processes contribute a novel way of treating input-dependent noise. In this respect, it is most closely related to Goldberg's *et al.* [20] approach and models the variance using a second Gaussian process in addition to a Gaussian process governing the noise-free output value. In contrast to Goldberg *et al.*, however, we do not use a time-consuming Markov chain Monte Carlo method to approximate the posterior noise variance but a fast most-likely-noise approach. This has the additional advantage that our approach fully stays within the Gaussian process framework so that more advanced Gaussian process techniques such as online learning, depending outputs, non-stationary covariance functions, and sparse approximations can easily be adapted.

III. GAUSSIAN BEAM PROCESSES

We pose the task of estimating $p(\mathbf{r}|\mathcal{A}, \mathbf{x})$ as a regression problem and model the function that maps beam angles α to range measurements r as a stochastic process. In other words, we regard the individual measurements r_i as a collection of random variables indexed by the respective beam angles α_i . By placing a Gaussian process prior over this function, we get a simple yet powerful model for likelihood estimation of range measurements as well as for prediction. Concretely, for mobile robot localization, we propose to build GBP models online for all robot pose hypotheses \mathbf{x} . The training set $\mathcal{D} = \{(\alpha_i, r_i)\}_{i=1}^n$ for learning such a model is simulated using ray casting operations relative to \mathbf{x} using a metric map of the environment. For certain applications, one needs to estimate $p(\mathbf{r}|\mathcal{A}, \mathcal{X})$, i.e. the distribution of range measurements for a

region \mathcal{X} in pose space. In this case, the training set \mathcal{D} is simply built by sampling poses \mathbf{x} from \mathcal{X} and simulating the corresponding range scans. In the following, we will derive the general model for d -dimensional angular indices α_i (e.g., $d = 1$ for planar sensing devices, $d = 2$ for 3D sensors).

Given a training set \mathcal{D} of range and bearing samples, we want to learn a model for the non-linear and noisy functional dependency $r_i = f(\alpha_i) + \epsilon_i$ with independent, normally distributed error terms ϵ_i . The idea of Gaussian processes is to view all target values r_i as jointly Gaussian distributed $p(r_1, \dots, r_n | \alpha_1, \dots, \alpha_n) \sim \mathcal{N}(\boldsymbol{\mu}, K)$ with a mean $\boldsymbol{\mu}$ and covariance matrix K .

The mean $\boldsymbol{\mu}$ is typically assumed $\mathbf{0}$ and K is defined by $k_{ij} := k(\alpha_i, \alpha_j) + \sigma_n^2 \delta_{ij}$, depending on a covariance function k and the global noise variance parameter σ_n . The covariance function represents the prior knowledge about the underlying function f and does not depend on the target values \mathbf{r} of \mathcal{D} . Common choices, that we also employ throughout this work, are the squared exponential covariance function

$$k_{SE}(\alpha_i, \alpha_j) = \sigma_f^2 \exp\left(-\frac{\Delta_{ij}^2}{2\ell^2}\right), \quad (1)$$

with $\Delta_{ij} = \|\alpha_i - \alpha_j\|$, which has a relatively strong smoothing effect, and a variant of the Matern type of covariance function $k_M(\alpha_i, \alpha_j) =$

$$\sigma_f^2 \left(1 + \frac{\sqrt{5}\Delta_{ij}}{\ell} + \frac{\sqrt{5}\Delta_{ij}^2}{3\ell^2}\right) \cdot \exp\left(-\frac{\sqrt{5}\Delta_{ij}}{\ell}\right). \quad (2)$$

These two covariance functions are called *stationary*, since they only depend on the distance Δ_{ij} between input locations α_i and α_j . In the definitions above, σ_f denotes the amplitude (or signal variance) and ℓ is the characteristic length-scale, see [21] for a detailed discussion. These parameters plus the global noise variance σ_n are called hyper-parameters of the process. They are typically denoted as $\Theta = (\sigma_f, \ell, \sigma_n)$. Since any set of samples from the process are jointly Gaussian distributed, predictions of m new range values $\mathbf{r}^* = (r_1^*, \dots, r_m^*)$, at given angles $\mathcal{A}^* = (\alpha_1^*, \dots, \alpha_m^*)$ can be performed by conditioning the $n + m$ -dimensional joint Gaussian on the known target values of the training set \mathcal{D} . This yields an m -dimensional predictive normal distribution $\mathbf{r}^* \sim \mathcal{N}(\boldsymbol{\mu}^*, \Sigma^*)$

$$\boldsymbol{\mu}^* = E(\mathbf{r}^*) = K^* (K + \sigma_n^2 I)^{-1} \mathbf{r} \quad (3)$$

$$\Sigma^* = V(\mathbf{r}^*) = K^{**} + \sigma_n^2 I - K^* (K + \sigma_n^2 I)^{-1} K^{*T} \quad (4)$$

with the covariance matrices $K \in \mathbb{R}^{n \times n}$, $K_{ij} = k(\alpha_i, \alpha_j)$, $K^* \in \mathbb{R}^{m \times n}$, $K_{ij}^* = k(\alpha_i^*, \alpha_j)$, and $K^{**} \in \mathbb{R}^{m \times m}$, $K_{ij}^{**} = k(\alpha_i^*, \alpha_j^*)$, and the training targets $\mathbf{r} \in \mathbb{R}^n$. The hyper-parameters of the Gaussian process can either be learned by maximizing the likelihood of the given data points or, for fully Bayesian treatment, can be integrated over using parameter-specific prior distributions. In this work, we adapt the hyper-parameters by maximizing the marginal likelihood of \mathcal{D} using the hybrid Monte-Carle approach described in [1].

So far, we have introduced the standard Gaussian processes framework for regression problems. In the following, we

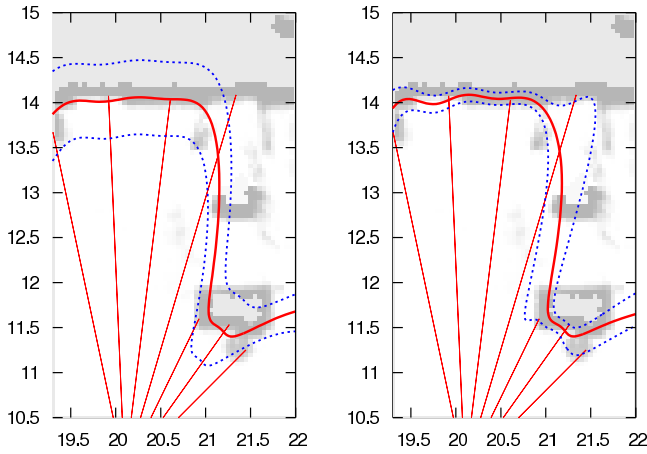


Fig. 3. The effect of modeling non-constant noise on a data set of range measurements simulated for the case of an uncertain sensor orientation ($\pm 5^\circ$). Standard Gaussian process regression (left) assumes constant noise for all bearing angles. Modeling heteroscedasticity (our model, on the right) yields lower predictive uncertainties at places with low expected noise levels such as the wall in front. The straight, red lines depict one possible range scan in this setting.

describe a novel way of treating input-dependent noise, which leads to more accurate models in our application domain.

A. Modeling Non-Constant Noise

Gaussian processes as introduced above assume a constant noise term, i.e., identically distributed error terms ϵ_i over the domain. For modeling range sensor measurements, however, the variance of range values in each beam direction is, along with its mean value, an important feature of the sought-after distribution of range measurements. To overcome this, we extended the standard Gaussian process framework to deal with heteroscedasticity, i.e., non-constant noise. Figure 3 illustrates the effect of this treatment on the predictive distribution for range values. The left diagram depicts the standard procedure that assumes a constant noise term for all bearings α . Our heteroscedastic treatment, depicted in the right diagram, achieves a significantly better fit to the data set while still not over-fitting to the individual samples.

To deal with the heteroscedasticity inherent in our problem domain, we basically follow the approach of Goldberg *et al.* [20], who condition a standard Gaussian processes \mathcal{G}_c on latent noise variables sampled from a separate noise process \mathcal{G}_n . Let $\mathbf{v} \in \mathbb{R}^n$ be such noise variances at the n given data points and $\mathbf{v}^* \in \mathbb{R}^m$ those for the m locations to be predicted, then the predictive distribution changes to

$$\boldsymbol{\mu}^* = K^* (K + K_v)^{-1} \mathbf{r}, \quad (5)$$

$$\Sigma^* = K^{**} + K_v^* - K^* (K + K_v)^{-1} K^{*T}, \quad (6)$$

where $K_v = \text{diag}(\mathbf{v})$ and $K_v^* = \text{diag}(\mathbf{v}^*)$. Now, as the noise variances \mathbf{v} and \mathbf{v}^* cannot be known a-priori, they have to be

integrated over for predicting \mathbf{r}^*

$$\begin{aligned} & p(\mathbf{r}^* | \mathcal{A}^*, \mathcal{D}) \\ &= \int \underbrace{p(\mathbf{r}^* | \mathcal{A}^*, \mathbf{v}, \mathbf{v}^*, \mathcal{D})}_{p_r} \cdot \underbrace{p(\mathbf{v}, \mathbf{v}^* | \mathcal{A}^*, \mathcal{D})}_{p_v} d\mathbf{v} d\mathbf{v}^*. \end{aligned} \quad (7)$$

Given the variances \mathbf{v} and \mathbf{v}^* , the prediction p_r in Eq. (7) is a Gaussian with mean and variance as discussed above. The problematic term is indeed p_v as it makes the integral difficult to handle analytically. Therefore, Goldberg *et al.* [20] proposed a Monte Carlo approximation that alternately samples from p_r and p_v to fit both curve and noise rates. The sampling is quite time consuming and the expectation can be approximated by the most likely noise levels $\tilde{\mathbf{v}}$ and $\tilde{\mathbf{v}}^*$. That is, we approximate the predictive distribution as

$$p(\mathbf{r}^* | \mathcal{A}^*, \mathcal{D}) \approx p(\mathbf{r}^* | \mathcal{A}^*, \tilde{\mathbf{v}}, \tilde{\mathbf{v}}^*, \mathcal{D}), \quad (8)$$

where $(\tilde{\mathbf{v}}, \tilde{\mathbf{v}}^*) = \arg \max_{(\tilde{\mathbf{v}}, \tilde{\mathbf{v}}^*)} p(\tilde{\mathbf{v}}, \tilde{\mathbf{v}}^* | \mathcal{A}^*, \mathcal{D})$. This will be a good approximation, if most of the probability mass of $p(\tilde{\mathbf{v}}, \tilde{\mathbf{v}}^* | \mathcal{A}^*, \mathcal{D})$ is concentrated around $(\tilde{\mathbf{v}}, \tilde{\mathbf{v}}^*)$. Moreover, the noise levels can be modeled using a standard Gaussian process. Thus, we have two interacting processes: \mathcal{G}_n predicts the noise levels and \mathcal{G}_c uses the predicted noise levels in (5) and (6). To learn the hyperparameters of both processes, we basically follow an alternating learning scheme in the spirit of the Expectation-Maximization algorithm: (1) fix the noise levels and learn \mathcal{G}_c using a standard maximum likelihood estimator; (2) fix \mathcal{G}_c , estimate the empirical noise levels of \mathcal{G}_c on the training data and estimated \mathcal{G}_n using them as target data. Initially, the noise levels are set to the empirical noise levels of a constant-noise Gaussian process induced on the training data.

As covariance functions, we use the Matern type as stated in Equation 2 for the range process and the squared exponential one for the noise process. This matches the intuition that the noise process should exhibit more smoothness than the range process, which was also supported by our experiments. This, however, is not a mandatory choice. With properly learned hyperparameters, using the squared exponential function for both processes yields a nearly as high performance in our application.

B. Evaluating the Joint Data Likelihood of Observations

For m new range measurements $\mathbf{z} = \{(\alpha_i, r_i)\}_{i=1}^m$ indexed by the beam orientations α_i , the model has to estimate the data likelihood $p(\mathbf{z} | \mathcal{D}, \Theta)$ given the training data \mathcal{D} and the learned covariance parameters Θ . We solve this by considering the predictive distribution for range measurements \mathbf{r}^* at the very same beam orientations $\alpha_1^*, \dots, \alpha_m^*$, which is an m -dimensional Gaussian distribution as defined by (5) and (6). As this predictive distribution is a multivariate Gaussian, we can directly calculate the observation likelihood for the data vector \mathbf{z} by evaluating the density function

$$\begin{aligned} p(\mathbf{z} | \boldsymbol{\mu}^*, \Sigma^*) &= \left[(2\pi)^{\frac{m}{2}} |\Sigma^*|^{\frac{1}{2}} \right]^{-1} \cdot \\ &\exp \left(-\frac{1}{2} (\mathbf{z} - \boldsymbol{\mu}^*)^T \Sigma^{*-1} (\mathbf{z} - \boldsymbol{\mu}^*) \right) \end{aligned} \quad (9)$$

or, in a more convenient form

$$\log p(\mathbf{z}|\boldsymbol{\mu}^*, \Sigma^*) = -\frac{1}{2}(\mathbf{z} - \boldsymbol{\mu}^*)^T \Sigma^{*-1} (\mathbf{z} - \boldsymbol{\mu}^*) - \frac{1}{2} \log |\Sigma^*| - \frac{m}{2} \log(2\pi). \quad (10)$$

C. Regression over Periodic Spaces

In our application, we have to account for the fact that our input vectors α_i are angular quantities rather than unconstrained real valued vectors. This means that an angular distance metric has to be incorporated into the covariance function to avoid discontinuities at $\pm\pi$. For the one dimensional case (for planar sensing devices), we use

$$\|\alpha, \beta\|_a := \begin{cases} |\alpha - \beta| & \text{if } |\alpha - \beta| \leq \pi \\ 2\pi - |\alpha - \beta| & \text{otherwise.} \end{cases} \quad (11)$$

Indeed, we also have to adapt the covariance functions themselves to the *periodic* structure of the input space. For example, a periodic variant of the squared exponential covariance function on the unit circle is

$$k(\alpha_i, \alpha_j) = \sigma_f^2 \sum_{p=-\infty}^{\infty} \exp\left(-\frac{|\alpha_i + 2\pi p - \alpha_j|^2}{2\ell^2}\right), \quad (12)$$

which takes infinitively many influences of a data point on itself into account. The squared exponential covariance function, however, has a strong locality for relevant values of σ_f^2 and ℓ . All summands with $|\alpha_i - \alpha_j| \geq 2\pi$ in Eq. (12) cannot even be represented using double precision, because their value is too close to zero. We can therefore safely ignore the periodicity in practice and only use the standard covariance function with the modified distance metric of Eq. (11).

D. Efficient Inference by Exploiting Locality

The covariance functions employed in this work are stationary, i.e., they assign small covariance values to those pairs of input points which lie far apart. With the given machine precision, this implies that the resulting covariance matrices are effectively band limited and only have non-zero entries close to the diagonal. This property can be exploited to speed up the computations by using optimized algorithms for sparse matrix operations. In this work, we used the UMFPAK package [22], an optimized solver for unsymmetric sparse linear systems, which resulted in significantly reduced computation times as shown in Figure 4. The run-times are given in seconds for a full iteration of simulating the scan, building the heteroscedastic model, and evaluating the observation likelihood for a given scan with 31 beams. The gain in speed depicted in this figure is due to the sparsity induced by the limitations of machine precision only. In addition to this, the covariances could be truncated actively to much tighter bounds before a notable loss of precision occurs.

IV. EXPERIMENTAL RESULTS

In this section, we will report on our experimental evaluation. The intention of this evaluation is to determine how well

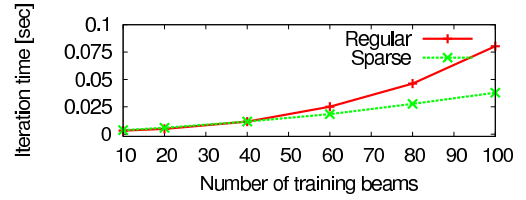


Fig. 4. The gain in speed due to sparse matrix calculations without a loss of precision. Exploiting sparsity reduces the iteration times drastically, especially for larger problem sizes.

the proposed GBP model performs compared to state-of-the-art probabilistic measurement models for laser range finders. To this aim, we applied our measurement model to one of the classical problems of robotics, mobile robot localization and tracking. We implemented our approach in C++. The platform used was a Pioneer PII DX8+ robot equipped with a laser range finder in a typical office environment. As the primary sensor is a planar sensing device, we only have to deal with one-dimensional bearing angles. To ensure a fair comparison, we independently optimized the parameters of all models using different data in all experiments.

We will proceed as follows. First, we briefly review the Monte Carlo localization scheme and discuss how the GBP model fits in there. Then, we present tracking and localization results with a real robot and, finally, we present results on simulated data to demonstrate the main benefits of our approach.

A. GBPs for Monte Carlo Localization

The task of mobile robot localization is to sequentially estimate the pose \mathbf{x} of a moving robot in its environment. The key idea of Monte Carlo localization (MCL) [23], which belongs to the class of particle filtering algorithms, is to maintain a sampled approximation of the probability density $p(\mathbf{x})$ of the robot's own location. This belief distribution is updated sequentially according to

$$p(\mathbf{x}^{[t]}|\mathbf{z}^{[1:t]}, \mathbf{u}^{[0:t-1]}) = \eta \cdot p(\mathbf{z}^{[t]}|\mathbf{x}^{[t]}) \cdot \int p(\mathbf{x}^{[t]}|\mathbf{u}^{[t-1]}, \mathbf{x}^{[t-1]}) \cdot p(\mathbf{x}^{[t-1]}|\mathbf{z}^{[1:t-1]}, \mathbf{u}^{[0:t-2]}) d\mathbf{x}^{[t-1]}. \quad (13)$$

Here, η is a normalization constant containing the prior observation likelihood $p(\mathbf{z}^{[t]})$, which is equal for the whole sample set and can therefore be neglected. The term $p(\mathbf{x}^{[t]}|\mathbf{u}^{[t-1]}, \mathbf{x}^{[t-1]})$ describes the probability that the robot is at position $\mathbf{x}^{[t]}$ given it executed the action $\mathbf{u}^{[t-1]}$ at position $\mathbf{x}^{[t-1]}$. Furthermore, $p(\mathbf{z}^{[t]}|\mathbf{x}^{[t]})$ denotes the probability of making observation $\mathbf{z}^{[t]}$ given the robot's current location is $\mathbf{x}^{[t]}$. The appropriate estimation of this quantity is the subject of this paper. Concretely, the update of the belief is realized by the following two alternating steps:

- 1) In the **prediction step**, we propagate each sample to a new location according to the robot's dynamics model $p(\mathbf{x}_t|\mathbf{u}_{t-1}, \mathbf{x}_{t-1})$ given the action \mathbf{u}_{t-1} executed since the previous update.

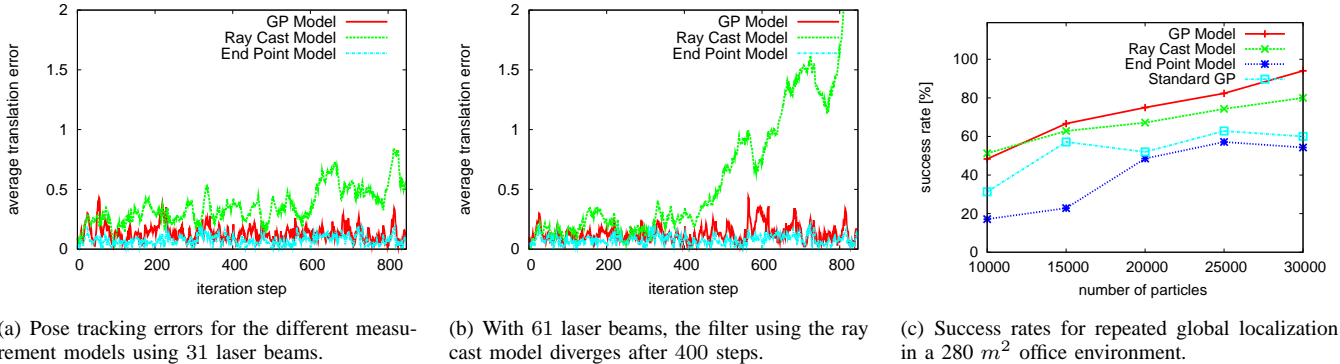


Fig. 5. Pose tracking and localization results with a real robot in an office environment using a 180 degrees field of view. Subfigures (a) and (b) give the tracking displacement error (y-axis) in meters for an increasing number of iterations. The errors are averaged over 25 runs on the same trajectory. Subfigure (c) shows the global localization performances.

- 2) In the **correction step**, the new observation \mathbf{z}_t is integrated into the sample set. This is done by adjusting the weight of each sample according to the likelihood $p(\mathbf{z}_t|\mathbf{x}_t)$ of sensing \mathbf{z}_t given the robot pose \mathbf{x}_t .

The measurement model $p(\mathbf{z}|\mathbf{x})$ plays a crucial role in the correction step of the particle filter. Typically, very peaked models require a large number of particles and induce a high risk of filter divergence. Even when the particles populate the state space densely, the likelihoods of an observation might differ by several orders of magnitude. As the particles are drawn proportionally to the importance weights, which themselves are calculated as the likelihood of \mathbf{z}_t given the pose \mathbf{x}_t of the corresponding particle, a minor difference in \mathbf{x}_t can already result in a large difference in likelihoods. This, in turn, would result in the depletion of such a particle in the re-sampling step. As a consequence, the “peakedness” of a measurement model should depend on the number of particles available and the size of the space to be covered. The ray cast model (RC) as well as the end point model (EP) have parameters for controlling their smoothness, which can be optimized for specific scenarios. In the following, we describe how GBPs can be applied to MCL and how the smoothness of the model can be defined in terms of an easy to interpret parameter.

As mentioned in Section III, we estimate $p(\mathbf{z}|\mathbf{x})$ by building a GBP model for the robot pose \mathbf{x} online and evaluating the data likelihood of \mathbf{z} according to Section III-B. For building the GBP model, we construct a training set \mathcal{D} of simulated range measurements relative to $\bar{\mathbf{x}} \sim \mathcal{N}(\mathbf{x}, \sigma_{\mathbf{x}})$ using an occupancy grid map of the environment. The random perturbations added to \mathbf{x} account for the desired smoothness of the model as motivated above. Indeed, the pose variance parameter $\sigma_{\mathbf{x}}$ introduced here, more naturally quantifies the level of regularization of GBPs compared to other models, as it is directly specified in the space of robot locations. Note that no sensory information is discarded at this point. For sufficiently high sampling densities, one could set $\sigma_{\mathbf{x}} = 0$ to get the fully peaked model.

The MCL measurement update step for the whole filter using GBPs can be summarized as follows:

Algorithm 1 GBPs-based Measurement Update for MCL

for all particles \mathbf{x} **do**

 Generate \mathcal{D} using ray casting in the given map at robot locations sampled from $\mathcal{N}(\mathbf{x}, \sigma_{\mathbf{x}})$.

 Build local GBPs using \mathcal{D} and the global covariance C .

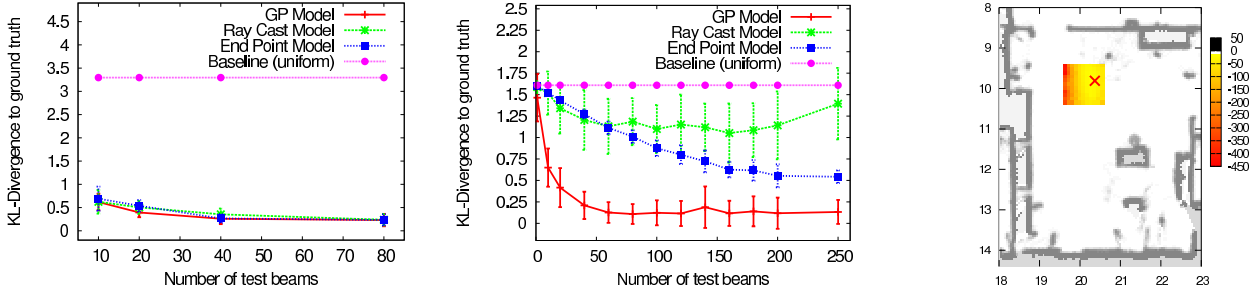
 Compute all $\log p(\mathbf{z}|GBP)$ and weight the particles.

end for

B. Results on Tracking and Localizing a Real Robot

To demonstrate the tracking and global localization performance of the GBP approach, we implemented Algorithm 1 and evaluated it using real data acquired with a Pioneer PII DX8+ robot equipped with a laser range scanner in a typical office environment. The experiments described here are designed to investigate how well our GBP approach performs in comparison to the widely used ray cast model and the end point model. While our approach is computationally more demanding than the alternative ones, it still runs close to real-time for mobile robot tracking. Here, a full iteration including scan simulation and model building takes approximately 0.011 seconds.

In the first set of experiments, we assess the position tracking performance of the MCL filter using the different measurement models. The robot started in the corridor of an office environment and traversed a path through all adjacent rooms. Figure 5(a) depicts the average localization error for this experiment with 31 laser beams. As can be seen, the GBP model and the end point model show similar, good localization performance and both outperform the ray cast model. When using more beams for the same task, the difference to the ray cast model gets even more pronounced, see Figure 5(b). Due to the ray cast model’s inability to deal with dependencies between beams, the risk of filter divergence increases with a growing number of beams used. In another experiment with 181 beams, the GBP model and the end point model showed



(a) Office room environment: All methods show similar performances and outperform the uniform model (pink) which assigns the same likelihood value to all grid cells. Lower KL-D values are better.

(b) Highly cluttered environment: The ray cast model (green) performs better than uniform (pink) but less well than the end point model (blue). Our GBP model (red) significantly outperforms the other methods in this environment.

(c) Observation log likelihood induced by the GBP model for a single simulated scan on a discretized pose space in a typical office room. It shows a convex shape and the true robot pose (cross) is in the area of maximal values.

Fig. 6. Localization performance for a mobile robot in terms of the Kullback-Leibler divergence (KLD), which measures the distance of the discretized pose belief distribution to the known ground truth (lower values are better). The experiments were simulated in an office (a) and in a cluttered environment (b). The KLD (y-axis) is shown for a varying number of laser beams (x-axis). The baseline model (uniform) assigns the same log likelihoods to all grid cells, i.e., it does not use any sensor information at all. Subfigure (c) shows a typical likelihood function induced by our Gaussian beam process (GBP) model.

a similar behavior as before. The ray cast model, however, diverged even earlier than with 61 beams.

In a second set of experiments we investigated the robustness of our GBP approach for global localization. Here, the task is to find the pose of a moving robot within an environment using a stream of wheel encoder and laser measurements. The environment used consists of a long corridor and 8 rooms containing chairs, tables and other pieces of furniture. In total, the map is 20 meters long and 14 meters wide. The results are summarized in Figure 5(c), which shows the number of successful localizations after 8 integrations of measurements for the three measurement models and for different numbers of particles used. In the experiment, we assumed that the localization was achieved when more than 95 percent of the particles differed in average at most 30 cm from the true location of the robot. As can be seen from the diagram, the GBP model performs slightly better than the ray cast model and both outperform the end point model.

C. Results on Simulated Data in a Static Setting

In the previous section, we have evaluated the measurement models in the standard way for mobile robots, i.e., we have evaluated their performances in real-world tracking and localization tasks. Although this is closest to the actual application of the models (and should therefore round off any other evaluation strategy), it has also one major drawback: several external factors influence the evaluation, such as the choice of filtering algorithm, the sampling and resampling strategies, and the order in which places are visited along a trajectory. To investigate the strengths and weaknesses of the measurement models independently from specific tracking algorithms, we ran a different set of experiments in a static setting. Here, we use the Kullback-Leibler divergence (KL-D) on a discretized pose space to measure how well the different models are able to reconstruct a pose distribution given just the corresponding laser measurements. More precisely, for each measurement model, we

- discretize the space of robot poses using a three dimensional grid (2D location and heading) and let each grid cell represent one pose hypothesis \mathcal{X}_i ,
- select a cell index t to contain the true robot pose $\mathbf{x}_t \in \mathcal{X}_t$,
- randomly draw m test poses within this cell t and simulate corresponding range measurement vectors \mathbf{z}_m using a given occupancy grid map.
- Now, we evaluate the observation likelihoods $p(\mathbf{z}_m|\mathcal{X}_i)$ for each grid cell and each test observation and sum up the individual observation likelihoods per cell.
- Finally, we normalize the whole likelihood grid and compute the KL-D $D_{KL} = \sum_i p(\mathbf{z}_m|\mathcal{X}_i) \cdot \log \left(\frac{p(\mathbf{z}_m|\mathcal{X}_i)}{\delta_{i=t}} \right)$ to a binary ground truth grid, where all likelihood mass is concentrated at the cell t , i.e., the true robot pose.

For computing the KL-D measure, we employ the standard trick of adding an extremely small value to each cell for dealing with empty cells. The specific choice of this value did not have a notable influence on the measure.

Figure 6(c) depicts such a likelihood grid for the GBP model as well as the true robot location in an office environment. It can be seen that the observation likelihood is nicely peaked around the true robot pose and that the GBP model yields a smooth likelihood function. The KL-D results for this room are given in Figure 6(a). The diagram shows that all three models achieve comparable good performances when recovering the pose distribution in this situation. Additionally, we plot the KL-D for the uniform model taken as a baseline, which assigns the same, constant likelihood value to all cells. In highly cluttered environments such as a laboratory room with many chairs and tables, however, the GBP model clearly outperforms the other two models. As shown in Figure 6(b), the KL-D is always significantly lower and decreases with a high number of laser beams. The ray cast model shows even an increasing KL-D with increasing numbers of laser beams due to its lack of smoothness. In both experiments, we used a relatively coarse



Fig. 7. In contrast to RC and GBP, the end point model (EP) cannot reliably distinguish between rooms A and B of the SDR building (left image). Since it only considers the end points of scans, it ignores the solid obstacle on the right-hand side of room B. Consequently, the resulting pose belief distribution shows a high KL-divergence to the ground truth (right) while RC and GBP achieve near optimal performances.

grid with grid cell areas of approximately $0.037m^2$.

Recall from the beginning that the end point model ignores the obstacles along the beam and replaces the ray casting operation of beam models by a nearest neighbor search. Figure 7 depicts a real situation in which this can lead to divergence of the localization process. To more quantitatively investigate this situation, we evaluated the different approaches in their capabilities to correctly classify laser scans simulated in room A as belonging to this room. As can be seen from the table on the right side of Figure 7, which contains the KL-D on a two cell grid (one cell for each room), the end point model produces a larger error than the ray cast model and the GBP model. This is mainly due to the fact that the end points of beams from room A also fit room B. The KL-D has been calculated for 100 simulated scans on a two cell grid, where each grid cell spans $1m^2$ in the center of room A respectively room B.

V. CONCLUSIONS

In this paper, we introduced Gaussian beam processes as a novel probabilistic measurement model for range finder sensors. The key idea of our approach is to view the measurement modeling task as a Bayesian regression problem and to solve it using Gaussian processes. As our experiments with real and simulated data demonstrate, Gaussian beam processes provide superior robustness compared to the ray cast model and the end point model by combining the advantages of both.

The model's ability to perform justified range predictions in arbitrary directions without having to consult a map is a promising starting point for future research. This might, for instance, allow to interpolate between measurements in settings where sensors only provide a sparse coverage of the environment.

ACKNOWLEDGMENTS

We would like to thank Andreas Klöckner for his invaluable help in optimizing the sparse matrix calculations as well as Andrew Howard for providing the SDR building data set through the Radish repository [24]. This work has been supported by the EC under contract number FP6-004250-CoSy, by the German Federal Ministry of Education and Research (BMBF) under contract number 01IME01F (project DESIRE), and by the German Research Foundation (DFG) within the Research Training Group 1103.

REFERENCES

- [1] C. Williams and C. Rasmussen, "Gaussian processes for regression," in *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*. MIT Press, 1995, pp. 514–520.
- [2] D. Fox, W. Burgard, and S. Thrun, "Markov localization for mobile robots in dynamic environments," *Journal of Artificial Intelligence Research*, vol. 11, pp. 391–427, 1999.
- [3] A. Petrovskaya, O. Khatib, S. Thrun, and A. Ng, "Bayesian estimation for autonomous object manipulation based on tactile sensors," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, Orlando, Florida, 2006.
- [4] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-gaussian tracking," in *IEEE Transactions on Signal Processing*, vol. 50, no. 2, February 2002, pp. 174–188.
- [5] B. Schiele and J. Crowley, "A comparison of position estimation techniques using occupancy grids," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 1994, pp. 1628–1634.
- [6] K. Konolige and K. Chou, "Markov localization using correlation," in *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, 1999.
- [7] S. Thrun, "A probabilistic online mapping algorithm for teams of mobile robots," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 335–363, 2001.
- [8] P. Pfaff, W. Burgard, and D. Fox, "Robust monte-carlo localization using adaptive likelihood models," in *European Robotics Symposium*, Palermo, Italy, 2006.
- [9] D. Fox, "Kld-sampling: Adaptive particle filters," in *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, 2001.
- [10] C. Kwok, D. Fox, and M. Meila, "Adaptive real-time particle filters for robot localization," in *Proc. of the IEEE International Conference on Robotics & Automation*, 2003.
- [11] R. Gutierrez-Osuna, J. Janet, and R. Luo, "Modeling of ultrasonic range sensors for localization of autonomous mobile robots," in *IEEE Transactions on Industrial Electronics*, vol. 45, no. 4, August 1998, pp. 654–662.
- [12] A. Schwaighofer, M. Grigoras, V. Tresp, and C. Hoffmann, "Gpps: A gaussian process positioning system for cellular networks," in *NIPS*, 2003.
- [13] A. Brooks, A. Makarenko, and B. Upcroft, "Gaussian process models for sensor-centric robot localisation," in *ICRA*, 2006.
- [14] B. Ferris, D. Haehnel, and D. Fox, "Gaussian processes for signal strength-based location estimation," in *Proceedings of Robotics: Science and Systems*, Philadelphia, USA, August 2006.
- [15] C. Plagemann, D. Fox, and W. Burgard, "Efficient failure detection on mobile robots using particle filters with gaussian process proposals," in *Proc. of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI)*, Hyderabad, India, 2007.
- [16] 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*, Philadelphia, USA, August 2006.
- [17] D. Grimes, R. Chalodhorn, and R. Rao, "Dynamic imitation in a humanoid robot through nonparametric probabilistic inference," in *Proceedings of Robotics: Science and Systems*, Philadelphia, USA, 2006.
- [18] C. Williams, "Prediction with gaussian processes: From linear regression to linear prediction and beyond," in *Learning and inference in graphical models*, M. I. Jordan, Ed. Kluwer Academic, 1998, pp. 599–621.
- [19] B. Schoelkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, "New support vector algorithms," *Neural Computation*, vol. 12, pp. 1207–1245, 2000.
- [20] P. Goldberg, C. Williams, and C. Bishop, "Regression with input-dependent noise: A gaussian process treatment," in *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, vol. 10, 1998.
- [21] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- [22] T. A. Davis, "A column pre-ordering strategy for the unsymmetric-pattern multifrontal method," *ACM Trans. Math. Softw.*, vol. 30, no. 2, pp. 165–195, 2004.
- [23] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1–2, pp. 99–141, 2000.
- [24] A. Howard and N. Roy, "The robotics data set repository (radish)," 2003. [Online]. Available: <http://radish.sourceforge.net/>