

# Adaptive Non-Stationary Kernel Regression for Terrain Modeling

Tobias Lang    Christian Plagemann    Wolfram Burgard

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

**Abstract**—Three-dimensional digital terrain models are of fundamental importance in many areas such as the geo-sciences and outdoor robotics. Accurate modeling requires the ability to deal with a varying data density and to balance smoothing against the preservation of discontinuities. The latter is particularly important for robotics applications, as discontinuities that arise, for example, at steps, stairs, or building walls are important features for path planning or terrain segmentation tasks. In this paper, we present an extension of the well-established Gaussian process regression technique, that utilizes non-stationary covariance functions to locally adapt to the structure of the terrain data. In this way, we achieve strong smoothing in flat areas and along edges and at the same time preserve edges and corners. The derived model yields predictive height distributions for arbitrary locations of the terrain and therefore allows us to fill gaps in the data and to perform conservative predictions in occluded areas.

## I. INTRODUCTION

The modeling of three-dimensional terrain has been widely studied across different research areas like the geo-sciences or robotics. Important applications in the latter case include mobile robotics for agriculture, search and rescue, or surveillance. In these domains, accurate and dense models of the three-dimensional structure of the environment enable the robot to estimate the traversability of locations, to plan its path to a goal location, or to localize itself using range sensor measurements. Building a digital terrain model means to transform a set of sensory inputs, typically a 3D point cloud or the raw range sensor readings, to a function mapping 2-dimensional pose coordinates to elevation values. While geological applications often operate on a larger spatial scale, where local terrain features can be neglected, autonomous robots greatly rely on distinct structural features like edges or corners to guide navigation, localization, or terrain segmentation. We therefore have two, at the first glance contradicting requirements for terrain models: First, raw sensory data needs to be smoothed in order to remove noise and to be able to perform elevation predictions at all locations and, second, discontinuities need to be preserved as they are important features for path planning, localization and object recognition.

In this paper, we present a novel terrain modeling approach based on an extended Gaussian process formulation. Our model uses non-stationary covariance functions as proposed by Paciorek *et al.* [7] to allow for local adaptation of the regression kernels to the underlying structure. This adaptation is achieved by iteratively fitting the local kernels to

the structure of the underlying function using local gradient features and the local marginal data likelihood (see Figure 1 for an illustration). Indeed, this idea is akin to adaptive image smoothing studied in computer vision, where the task is to achieve de-noising of an image without reducing the contrast of edges and corners [14, 6]. Although these approaches from the computer vision literature are not specifically designed for dealing with a varying density of data points or with potential gaps to fill, they nevertheless served as an inspiration for our kernel adaptation approach.

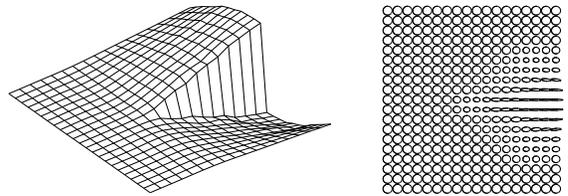


Fig. 1. A hard, synthetic regression problem (left). The continuous regions should be smoothed without removing the strong edge feature. Our approach achieves this by adapting local kernels to the terrain data (right).

The paper is structured as follows. We first discuss related work in the next section. In Section III, we formalize the terrain modeling problem using Gaussian processes and introduce our approach to non-stationary adaptive regression. Section IV presents our experimental results on real and simulated terrain data sets.

## II. RELATED WORK

A broad overview over methods used for modeling terrain data is given by Hugentorp [5]. Elevation maps have been used as an efficient data structure for representing dense terrain data [1, 8] and have later been extended to multi-level probabilistic surface maps [16]. Früh *et al.* [4] present an approach to filling local gaps in 3D models based on local linear interpolation. As their approach has yielded promising results in city mapping applications, we compare its modeling accuracy to our approach in Section IV.

Gaussian processes (GPs) have a long tradition in the geo-sciences and statistics literature [11]. Classical approaches for dealing with non-stationarity include input-space warping [12, 13] and hierarchical modeling using local kernels [7]. The latter approach provides the general framework for this work.

Recently, GPs have become popular in robotics, e.g., for learning measurement models [2] or model-based failure detection [9]. To deal with varying target function properties in the context of perception problems, Williams [17] uses mixtures of GPs for segmenting foreground and background in images in order to extract disparity information from binocular stereo images. Rasmussen and Ghahramani [10] extend ideas of Tresp [15] and present an infinite mixture of experts model where the individual experts are made up from different GP models. A gating network assigns probabilities to the different expert models based completely on the input. Discontinuities in wind fields have been dealt with by Cornford *et al.* [3]. They place auxiliary GPs along the edge on both sides of the discontinuity. These are then used to learn GPs representing the process on either side of the discontinuity. In contrast to our work, they assume a parameterized segmentation of the input space, which appears to be disadvantageous in situations such as depicted in Figure 1 and on real-world terrain data sets.

The problem of adapting to local structure has also been studied in the computer vision community. Taketa *et al.* [14] perform non-parametric kernel regression on images. They adapt kernels according to observed image intensities. Their adaptation rule is thus based on a nonlinear combination of both spatial and intensity distance of all data points in the local neighborhood. Based on singular value decompositions of intensity gradient matrices, they determine kernel modifications. Middendorf and Nagel [6] propose an alternative kernel adaptation algorithm. They use estimates of gray value structure tensors to adapt smoothing kernels to gray value images.

### III. DIGITAL TERRAIN MODELING

Data for building 3-dimensional models of an environment can be acquired from various sources. In robotics, laser range finders are popular sensors as they provide precise, high-frequency measurements at a high spatial resolution. Other sensors include on-board cameras, which are chosen because of their low weight and costs, or satellite imagery, which covers larger areas, e.g., for guiding unmanned aerial vehicles (UAVs) or autonomous cars. After various preprocessing steps, the raw measurements are typically represented as 3D point clouds or are transformed into a 3D occupancy grid or elevation map [1]. In this work, we introduce a technique for constructing continuous, probabilistic elevation map models from data points, that yield predictive distributions for terrain elevations at arbitrary input locations.

The terrain modeling problem can be formalized as follows. Given a set  $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$  of  $n$  location samples  $\mathbf{x}_i \in \mathbb{R}^2$  and the corresponding terrain elevations  $y_i \in \mathbb{R}$ , the task is to build a model for  $p(y^* | \mathbf{x}^*, \mathcal{D})$ , i.e., the predictive distribution of elevations  $y^*$  at new input locations  $\mathbf{x}^*$ . This modeling task is a hard one for several reasons. First, sensor measurements are inherently affected by noise, which an intelligent model should be able to reduce. Second, the distribution of available data points is typically far from uniform. For example, the proximity of the sensor location is usually more densely

sampled than areas farther away. Third, small gaps in the data should be filled with high confidence while more sparsely sampled locations should result in higher predictive uncertainties. To illustrate the last point, consider an autonomous vehicle navigating in off road terrain. Without filling small gaps, even single missing measurements may lead to the perception of an un-traversable obstacle and consequently the planned path might differ significantly from the optimal one. On the other hand, the system should be aware of the increased uncertainty when filling larger gaps to avoid overconfidence at these locations. As a last non-trivial requirement, the model should preserve structural elements like edges and corners as they are important features for various applications including path planning or object recognition.

In this paper, we propose a model to accommodate for all of the above-mentioned requirements. We build on the well-established framework of Gaussian processes, which is a non-parametric Bayesian approach to the regression problem. To deal with the preservation of structural features like edges and corners, we employ non-stationary covariance functions as introduced by Paciorek and Schervish [7] and present a novel approach to local kernel adaptation based on gradient features and the local marginal data likelihood.

In the following, we restate the standard Gaussian process approach to non-parametric regression before we introduce our extensions to local kernel adaptation.

#### A. Gaussian Process Regression

As stated in the previous section, the terrain modeling task is to derive a model for  $p(y^* | \mathbf{x}^*, \mathcal{D})$ , which is the predictive distribution of terrain elevations  $y^*$ , called *targets*, at input locations  $\mathbf{x}^*$ , given a training set  $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$  of elevation samples. The idea of Gaussian processes (GPs) is to view any finite set of samples  $y_i$  from the sought after distribution as being jointly normally distributed,

$$p(y_1, \dots, y_n | \mathbf{x}_1, \dots, \mathbf{x}_n) \sim \mathcal{N}(\boldsymbol{\mu}, K), \quad (1)$$

with mean  $\boldsymbol{\mu} \in \mathbb{R}^n$  and covariance matrix  $K$ .  $\boldsymbol{\mu}$  is typically assumed  $\mathbf{0}$  and  $K$  is specified in terms of a parametric covariance function  $k$  and a global noise variance parameter  $\sigma_n$ ,  $K_{ij} := k(\mathbf{x}_i, \mathbf{x}_j) + \sigma_n^2 \delta_{ij}$ . The covariance function  $k$  represents the prior knowledge about the target distribution and does not depend on the target values  $y_i$  of  $\mathcal{D}$ . A common choice is the squared exponential covariance function

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp\left(-\frac{1}{2} \sum_{k=1}^2 \frac{(\mathbf{x}_{i,k} - \mathbf{x}_{j,k})^2}{\ell_k}\right), \quad (2)$$

where  $\sigma_f$  denotes the amplitude (or signal variance) and  $\ell_k$  are the characteristic length-scales of the individual dimensions (see [11]). These parameters plus the global noise variance  $\sigma_n$  are called hyperparameters of the process. They are typically denoted as  $\boldsymbol{\Theta} = (\sigma_f, \boldsymbol{\ell}, \sigma_n)$ . Since any set of samples from the process is jointly Gaussian distributed, the prediction of a new target value  $y^*$  at a given location  $\mathbf{x}^*$  can be performed by conditioning the  $n + 1$ -dimensional joint Gaussian on the

known target values of the training set  $\mathcal{D}$ . This yields a predictive normal distribution  $y^* \sim \mathcal{N}(\mu^*, v^*)$  defined by

$$\mu^* = E(y^*) = \mathbf{k}^T (K + \sigma_n^2 I)^{-1} \mathbf{y}, \quad (3)$$

$$v^* = V(y^*) = k^* + \sigma_n^2 - \mathbf{k}^T (K + \sigma_n^2 I)^{-1} \mathbf{k}, \quad (4)$$

with  $K \in \mathbb{R}^{n \times n}$ ,  $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ ,  $\mathbf{k} \in \mathbb{R}^n$ ,  $k_j = k(\mathbf{x}^*, \mathbf{x}_j)$ ,  $k^* = k(\mathbf{x}^*, \mathbf{x}^*) \in \mathbb{R}$ , and the training targets  $\mathbf{y} \in \mathbb{R}^n$ . Learning in the Gaussian process framework means finding the parameters  $\Theta$  of the covariance function  $k$ . Throughout this work we use a conjugate gradient based algorithm [11] that fixes the parameters by optimizing the marginal data likelihood of the given training data set. Alternatively, the parameters could be integrated over using parameter-specific prior distributions, which results in a fully Bayesian model but which is also computationally more demanding as one has to employ Markov-Chain Monte Carlo sampling for approximating the intractable integral.

The standard model introduced so far already accounts for three of the requirements discussed in the previous section, namely de-noising, dealing with non-uniform data densities, and providing predictive uncertainties. As a major drawback, however, by using the stationary covariance function of Equation (2), which depends only on the *differences* between input locations, one basically assumes the same covariance structure on the whole input space. In practice, this significantly weakens important features like edges or corners. The left diagram of Figure 1 depicts a synthetic data-set which contains homogenous regions which should be smoothed, but also a sharp edge that has to be preserved. Our model, which is detailed in the next section, addresses this problem by adapting a non-stationary covariance function to the local terrain properties.

### B. Non-Stationary Covariance Functions

Most Gaussian process based approaches found in the literature use stationary covariance functions that depend on the difference between input locations  $\mathbf{x} - \mathbf{x}'$  rather than on the absolute values  $\mathbf{x}$  and  $\mathbf{x}'$ . A powerful model for building non-stationary covariance functions from arbitrary stationary ones has been proposed by Paciorek and Schervish [7]. For the Gaussian kernel, their non-stationary covariance function takes the simple form

$$k(\mathbf{x}_i, \mathbf{x}_j) = |\Sigma_i|^{\frac{1}{4}} |\Sigma_j|^{\frac{1}{4}} \left| \frac{\Sigma_i + \Sigma_j}{2} \right|^{-\frac{1}{2}} \exp \left[ -(\mathbf{x}_i - \mathbf{x}_j)^T \left( \frac{\Sigma_i + \Sigma_j}{2} \right)^{-1} (\mathbf{x}_i - \mathbf{x}_j) \right], \quad (5)$$

where each input location  $\mathbf{x}'$  is assigned an individual Gaussian kernel matrix  $\Sigma'$  and the covariance between two targets  $y_i$  and  $y_j$  is calculated by averaging between the two individual kernels at the input locations  $\mathbf{x}_i$  and  $\mathbf{x}_j$ . In this way, the local characteristics at both locations influence the modeled covariance of the corresponding target values. In this model, each kernel matrix  $\Sigma_i$  is internally represented by its eigenvectors

and eigenvalues. Paciorek and Schervish build a hierarchical model by placing additional Gaussian process priors on these kernel parameters and solve the integration using Markov-Chain Monte Carlo sampling. While the model presented in [7] provides a flexible and general framework, it is, as also noted by the authors, computationally demanding and clearly not feasible for the real world terrain data sets that we are aiming for in this work. As a consequence, we propose to model the kernel matrices in Equation (5) as independent random variables that are initialized with the learned kernel of the corresponding stationary model and then iteratively adapted to the local structure of the given terrain data. Concretely, we assign to every input location  $\mathbf{x}_i$  from the training set  $\mathcal{D}$  a local kernel matrix  $\Sigma_i$ , which in turn is represented by one orientation parameter and two scale parameters for the length of the axes. Given these parameters, the evaluation of Equation (5) is straightforward. In the following section, we will discuss in detail, how the kernel matrices  $\Sigma_i$  can be adapted to the local structure of the terrain.

### C. Local Kernel Adaptation

The problem of adapting smoothing kernels to local structure has been well studied in the computer vision community. It is therefore not surprising that, although image processing algorithms are typically restricted to dense and uniformly distributed data, we can use findings from that field as an inspiration for our terrain adaptation task. Indeed, Mitterdorf and Nagel [6] present a technique for iterative kernel adaptation in the context of optical flow estimation in image sequences. Their approach builds on the concept of the so called grey-value structure tensor (GST), which captures the local structure of an image or image sequence by building the locally weighted outer product of grey-value gradients in the neighborhood of the given image location. Analogously to their work, we define the elevation structure tensor (EST) for a given location  $\mathbf{x}_i$  as

$$EST(\mathbf{x}_i) := \overline{\nabla y (\nabla y)^T}(\mathbf{x}_i), \quad (6)$$

where  $y(\mathbf{x})$  denotes the terrain elevation at a location  $\mathbf{x}$  and  $\overline{\cdot}$  stands for the operator that builds a locally weighted average of its argument according to the kernel  $\Sigma_i$ . For two-dimensional  $\mathbf{x}_i$ , Equation (6) calculates the locally weighted average of the outer product of  $\nabla y = (\frac{\partial y}{\partial x_1}, \frac{\partial y}{\partial x_2})^T$ . This local elevation derivative can be estimated directly from the raw elevation samples in the neighborhood of the given input location  $\mathbf{x}_i$ . We cope with the noise stemming from the raw data by averaging over the terrain gradients in the local neighborhood.

Equation (6) yields a tensor, representable as a  $2 \times 2$  real-valued matrix, which describes how the terrain elevation changes in the local neighborhood of location  $\mathbf{x}_i$ . To get an intuition, what  $EST(\mathbf{x}_i)$  encodes and how this can guide the adaptation of the local kernel  $\Sigma_i$ , consider the following situations. Let  $\lambda_1$  and  $\lambda_2$  denote the eigenvalues of  $EST(\mathbf{x}_i)$  and  $\beta$  be the orientation angle of the first eigenvector. If  $\mathbf{x}_i$  is located in a flat part of the terrain, the elevation gradients

$\nabla y$  are small in the neighborhood of  $\mathbf{x}_i$ . This results in two equally small eigenvalues of  $EST(\mathbf{x}_i)$ . In contrast, if  $\mathbf{x}_i$  was located in an ascending part of the terrain, the first eigenvalue of  $EST(\mathbf{x}_i)$  would be clearly greater than the second one and the orientation  $\beta$  would point towards the strongest ascent.

Intuitively and as discussed in more detail by Middendorf and Nagel [6], the kernel  $\Sigma_i$  describing the extent of the local environment of  $\mathbf{x}_i$  should be set to the inverse of  $EST(\mathbf{x}_i)$ . In this way, flat areas are populated by large, isotropic kernels, while sharp edges have long, thin kernels oriented along the edge directions. Corner structures, having strong elevation gradients in all dimensions, result in relatively small local kernels. To prevent unrealistically large kernels, Middendorf and Nagel describe how this inversion can be bounded to yield kernels, whose standard deviations lie between given values  $\sigma_{min}$  and  $\sigma_{max}$ . Based on their findings, we give three concrete local adaptation rules that have been compared in our experimental evaluation. To simplify notation, we introduce  $\bar{\lambda}_k = \lambda_k / (\lambda_1 + \lambda_2)$ ,  $k = 1, 2$  and the re-parameterization

$$\Sigma_i = R^{-T} \begin{pmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{pmatrix} R^{-1} \quad (7)$$

where  $\alpha_1$  and  $\alpha_2$  scale in orthogonal directions and  $R$  is a rotation matrix specified by the orientation angle  $\theta$ .

- 1) *Direct Inverse Adaptation*:  $\Sigma_i = EST(\mathbf{x}_i)^{-1}$
- 2) *Bounded Linear Adaptation*:

$$\alpha_k = \bar{\lambda}_k \sigma_{min}^2 + (1 - \bar{\lambda}_k) \sigma_{max}^2, \quad k = 1, 2$$

- 3) *Bounded Inverse Adaptation*:

$$\alpha_k = \frac{\sigma_{max}^2 \sigma_{min}^2}{\bar{\lambda}_k \sigma_{max}^2 + (1 - \bar{\lambda}_k) \sigma_{min}^2}, \quad k = 1, 2$$

The two *bounded* adaptation procedures prevent unrealistically small and large kernels. The *Bounded Inverse* strongly favors the larger eigenvalue dimension and produces more pronounced kernels (larger difference between semiaxes) while the *Bounded Linear* Linear tends to produce more balanced and larger kernels. This is why *Bounded Linear* performs better in the presence of sparse data as it is less vulnerable to overfitting. In this work, the bounds  $\sigma_{min}$  and  $\sigma_{max}$  are estimated empirically. We are currently working on determining optimal values with respect to the marginal data likelihood.

So far, we have described how to perform one local adaptation step for an arbitrary kernel  $\Sigma_i$ . As the complete learning and adaptation procedure, which is summarized in Algorithm 1, we propose to assign to each input location  $\mathbf{x}_i$  of the training set  $\mathcal{D}$  a kernel matrix  $\Sigma_i$ , which is initialized with a global parameter vector  $\Theta$ , that in turn has been learned using standard GP learning with the corresponding stationary covariance function. The local kernels are then iteratively adapted to the elevation structure of the given terrain data set until their parameters have converged. To quickly adapt the kernels at locations where the regression error is high (relative to the given training data set), we propose to make the adaptation speed for each  $\Sigma_i$  dependent on the local data fit  $df(\mathbf{x}_i)$ , which is the normalized observation likelihood of the

corresponding  $y_i$  from the training set relative to the current predictive distribution (see Equation (III-A)), and the kernel complexity approximated as  $c_i = 1/|\Sigma_i|$ . Both quantities are used to form a learning rate parameter calculated by means of a modified sigmoid function,  $\eta_i = \text{sigmoid}(-df(\mathbf{x}_i) \cdot c_i; \delta)$ , where the additional parameters  $\delta$  are determined empirically. Intuitively, we get a high adaptation speed when the data-fit relative to the kernel size is small. Algorithm 1 summarizes the adaptation procedure.

---

#### Algorithm 1 Local Kernel Adaptation

---

Learn global parameters  $\Theta$  for the stationary squared exponential covariance function.

Initialize all local kernels  $\Sigma_i$  with  $\Theta$ .

**while** not converged **do**

**for** all  $\Sigma_i$  **do**

    Estimate the local learning rate  $\eta_i$

    Estimate  $EST(\mathbf{x}_i)$  according to  $\Sigma_i$

$\Sigma_i^* \leftarrow \text{ADAPT}(EST(\mathbf{x}_i))$

$\Sigma_i \leftarrow \eta_i \Sigma_i^* + (1 - \eta_i) \Sigma_i$

**end for**

**end while**

---

## IV. EXPERIMENTAL EVALUATION

The goals of the experimental evaluation presented in this section are (a) to show that our terrain modeling approach is indeed applicable to real data sets, (b) that our model is able to remove noise while at the same time preserving important structural features, and (c) that our model yields more accurate and robust elevation predictions at sparsely sampled input locations than an alternative approach to this problem.

As an evaluation metric, we use the mean squared error  $MSE(\mathcal{X}) = \frac{1}{m} \sum_{i=1}^m (y_i - y_i^*)^2$  of predicted elevations  $y_i^*$  relative to ground truth elevations  $y_i$  on a set of input locations  $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^m$ .

### A. Evaluation on Artificial Terrain Data

The first set of experiments was designed to quantify the benefits of local kernel adaptation and to compare the three different adaptation rules. As a test scenario, we took the artificial terrain data set depicted in Figure 2 consisting of 441 data points, which contains uniform regions as well as sharp edges and corners, which are hard to adapt to locally. Note, for example, that the edge between the lowest and the second lowest plateau has a curvature and that three different height levels can be found in the local neighborhood of the corner in the middle of the diagram. We set  $\sigma_{min} = 0.001$  and  $\sigma_{max} = 5.0$  for the bounded adaptation rules.

To generate training data sets for the different experiments reported on here, we added white noise of a varying standard deviation  $\sigma$  to the true terrain elevations and randomly removed a portion of the samples to be able to assess the model's predictive abilities.

Figure 4 visualizes a complete adaptation process for the case of a data set generated using a noise rate of  $\sigma = 0.3$ .

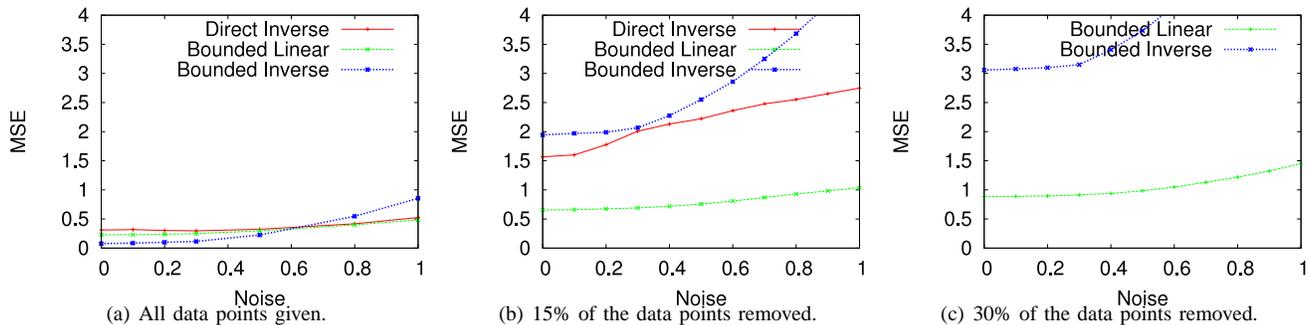


Fig. 3. Prediction accuracy for the scenario depicted in Figure 4 with (a) all data points available, (b) 15% of the data-points randomly removed and (c) 30% randomly removed. Each figure plots the mean squared error of elevation predictions for a varying level of added white noise. The values are averaged over 10 independent runs per configuration. (In the case of (c), the error of *Direct Inverse* was always greater than 4.0).

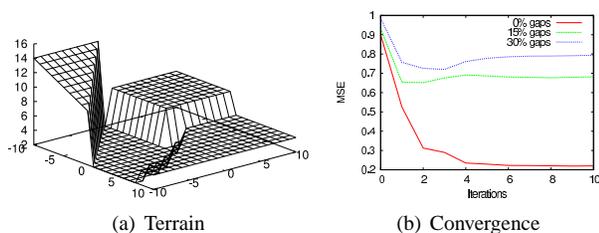


Fig. 2. An artificial terrain data set used in the experimental evaluation, that exhibits several local features that are hard to adapt to (a). Test data sets are generated by adding white noise and randomly removing a portion of the data points. The mean squared error (MSE) of predicted elevations converges with an increasing number of adaptation steps (b). Iteration 0 gives the MSE for the learned standard GP. Values are averaged over ten independent runs.

On average, a single iteration per run took 44 seconds on this data-set using a PC with a 2.8 GHz CPU and 2 GB of RAM. Figures 4(c)-4(f) show the results of standard GP regression which places the same kernels at all input locations. While this leads to good smoothing performance in homogeneous regions, the discontinuities within the map are also smoothed as can be seen from the absolute errors in the third column. Consequently, those locations get assigned a high learning rate, see right column, used for local kernel adaption.

The first adaptation step leads to the results depicted in Figures 4(g)-4(j). It is clearly visible, that the steps and corners are now better represented by the regression model. This has been achieved by adapting the kernels to the local structure, see the first column of this row. Note, how the kernel sizes and orientations reflect the corresponding terrain properties. Kernels are oriented along discontinuities and are small in areas of strongly varying elevation. In contrast, they have been kept relatively large in homogeneous regions. After three iterations, the regression model has adapted to the discontinuities accurately while still de-noising the homogeneous regions (Figures 4(k)-4(n)). Note, that after this iteration, the local learning rates have all settled at low values.

Figure 2 gives the convergence behavior of our approach using the *Bounded Linear* adaptation rule in terms of the

mean squared prediction error for different amounts of points removed from the noisy data set. After at most 6 iterations, the errors have settled close to their final value.

In a different set of experiments, we investigated the prediction performance of our approach for all three adaptation rules presented in Section III-C. For this experiment, we added white noise of a varying noise level to the artificial terrain given in Figure 2. The diagrams in Figure 3 give the results for different amounts of points removed from the noisy data set. When no points are removed from the test set, the *Bounded Inverse* adaptation rule performs best for small noise values. For large noise values, *Bounded Linear* and *Direct Inverse* achieve better results. In the case of 15% and 30% data points removed, *Direct Inverse* and *Bounded Inverse* are not competitive. In contrast, *Bounded Linear* still achieves very good results for all noise levels.

Thus, *Bounded Linear* produces reliable predictions for all tested noise rates and data densities. This finding was supported by experiments on other real data sets not presented here.

### B. Evaluation on Real Terrain Data

In order to demonstrate the usefulness of our approach on real data sets, we acquired a set of 3D scans of a scene using a mobile robot equipped with a laser range finder, see Figure 5(a). We compared our prediction results to an approach from the robotics literature [4] that has been applied successfully to the problem of 3-dimensionally mapping urban areas. We employed the *Bounded Linear* adaptation procedure for our learning algorithm where we set  $\sigma_{min} = 0.25$  and  $\sigma_{max} = 4.0$ . Figure 5 gives the results of this experiment. An obstacle, in this case a person, is placed in front of the robot and thus occludes the sloped terrain behind.

We evaluated our approach for the situation depicted in the figure as well as for three similar ones and compared its prediction accuracy to the approach of Früh *et al.* [4], who perform horizontal linear interpolation orthogonally to the robot's view. These scenarios used are actually rather easy ones for [4], as the large gaps can all be filled orthogonally to

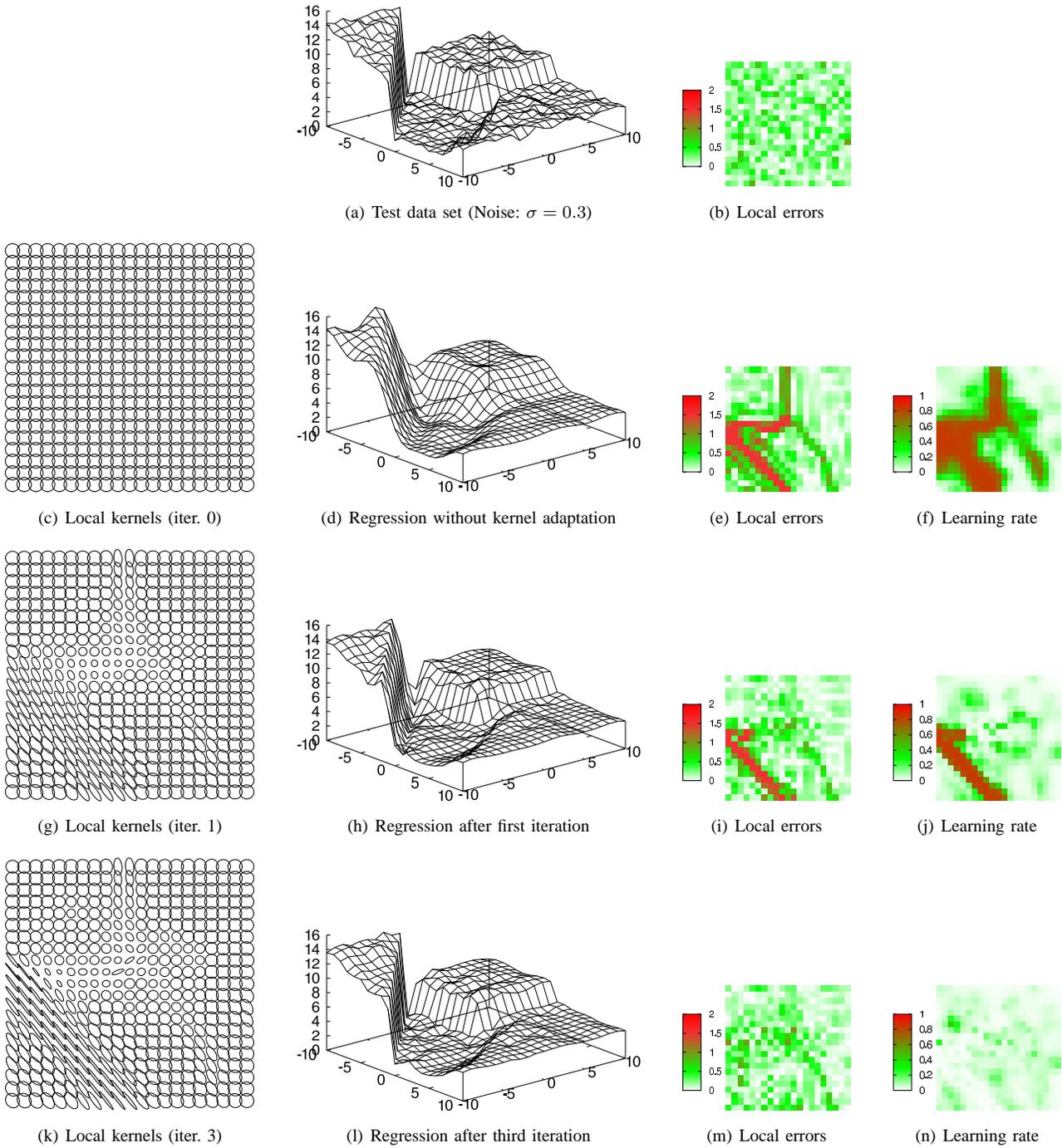


Fig. 4. The local kernel adaptation process on an artificial terrain data set: the original data set, depicted in Figure 2, exhibits several local features that are hard to adapt to. The test data set (a) was generated by adding white noise, resulting in the errors shown in (b). The second row of diagrams gives information about the initialization state of our adaptation process, i.e. the results of standard GP learning and regression. The following two rows depict the results of our approach after the first and after the third adaptation iteration respectively. In the first column of this figure, we visualize the kernel dimensions and orientations after the corresponding iteration. The second column depicts the predicted means of the regression. The third column gives the absolute errors to the known ground truth elevations and the right-most column gives the resulting learning rates  $\eta_i$  for the next adaptation step resulting from the estimated data likelihoods.

the robot’s view, which is not the case in general. To estimate the kernels at unseen locations, we built a weighted average over the local neighborhood with an isotropic two-dimensional Gaussian with a standard deviation of 3 which we had found to produce the best results. Table I gives the results. In all four cases, our approach achieved higher prediction accuracies, reducing the errors by 30% to 70%. Figure 5(b) depicts the predictions of our approach in one of the situations. In contrast to Früh *et al.*, our model is able to also give the predictive uncertainties. These variances are largest in the center of the occluded area as can be seen in Figure 5(c).

In a second real-world experiment illustrated in Figure 6, we investigated the ability of our terrain model approach to preserve and predict sharp discontinuities in real terrain data. We positioned the robot in front of a rectangular stone block such that the straight edges of the block run diagonally to the robot’s line of view. A person stood in between the robot and the block, thereby occluding parts of the block and of the area in front of it. This scenario is depicted in 6(a). The task is to recover the linear structure of the discontinuity and fill the occluded area consistent with the surrounding terrain elevation levels. The adaptation procedure converged already after two iterations. The learned kernel structure, illustrated in Figure 6(c), enables the model to correctly represent the stone blocks as can be seen from the predicted elevations visualized in 6(d). This figure also illustrates the uncertainties of these predictions, corresponding to the variances of the predictive distributions, by means of two contour lines. This indicates that a mobile robot would be relatively certain about the block structure within the gap although not having observed it directly. In contrast, it would be aware that it cannot rely upon its terrain model in the occluded areas beyond the blocks: there are no observations within a reasonable distance and thus, the predictive variances are large.

To show that our approach is applicable to large, real-world problems, we have tested it on a large data-set recorded at the University of Freiburg campus<sup>1</sup>. The raw terrain data was preprocessed, corrected, and then represented in a multi-level surface map with a cell size of 10cm × 10cm. The scanned area spans approximately 299 by 147 meters. For simplicity, we only considered the lowest data-points per location, i.e., we removed overhanging structures like tree tops or ceilings. The resulting test set consists of 531,920 data-points. To speed up computations, we split this map into 542 overlapping sub-maps. This is possible without loss of accuracy as we can assume compact support for the local kernels involved in our calculations (as the kernel sizes in our model are bounded). We randomly removed 20% of the data-points per sub-map. A full run over the complete data-set took about 50 hours. Note that the computational complexity can be reduced substantially by exploiting the sparsity of our model (due to the bounded kernels) and by introducing additional sparsity using approximative methods, e.g., sparse GPs. Table II gives

Scenario	Linear Interp. [4]	Adapted GP	Improvement
1 (Fig. 5)	0.116	0.060	48.3%
2	0.058	0.040	31.0%
3	0.074	0.023	69.9%
4	0.079	0.038	51.9%

TABLE I  
PREDICTION PERFORMANCE IN TERMS OF MSE RELATIVE TO A SECOND, NOT OCCLUDED SCAN.

Adaptation procedure	MSE
<i>Standard GP</i>	0.071
<i>Direct Inverse</i>	0.103
<i>Bounded Linear</i>	0.062
<i>Bounded Inverse</i>	0.059

TABLE II  
PREDICTION PERFORMANCE ON A LARGE CAMPUS ENVIRONMENT.

the results of this experiment for the different adaptation rules. The *Bounded Linear* and the *Bounded Inverse* adaptation procedures outperform the *Standard GP* model where kernels are not adapted, while *Direct Inverse* is not competitive. Together with the results of the other experiments, this leads to the conclusion that *Bounded Linear* is an adequate choice as an adaptation rule in synthetic and real-world scenarios.

## V. CONCLUSIONS

In this paper, we propose an adaptive terrain modeling approach that balances smoothing against the preservation of structural features. Our method uses Gaussian processes with non-stationary covariance functions to locally adapt to the structure of the terrain data. In experiments on synthetic and real data, we demonstrated that our adaptation procedure produces reliable predictions in the presence of noise and is able to fill gaps of different sizes. Compared to a state-of-the-art approach from the robotics literature we achieve a prediction error reduced by approximately 30%-70%.

In the future, we intend to evaluate our approach in online path planning applications for mobile robots. Since our approach retrieves terrain properties in terms of kernels, its application to terrain segmentation is promising. Another direction of further research are SLAM techniques where the trajectory of the robot is also unknown and the model has to be updated sequentially. We also intend to evaluate our approach on typical test cases in computer vision and to compare it with the algorithms of this community. Finally, we work on an analytical derivation for optimal kernels based solely on data likelihoods and model complexity.

## VI. ACKNOWLEDGMENTS

The authors would like to thank Kristian Kersting for the stimulating discussion as well as Rudolph Triebel and Patrick Pfaff for providing the campus data-set and their source code for multi-level surface maps. This work has been supported by the EC under contract number FP6-004250-CoSy and by the German Federal Ministry of Education and Research (BMBF) under contract number 01IME01F (project DESIRE).

<sup>1</sup>Additional material for the campus experiment can be found at <http://www.informatik.uni-freiburg.de/plagem/rss07terReg>

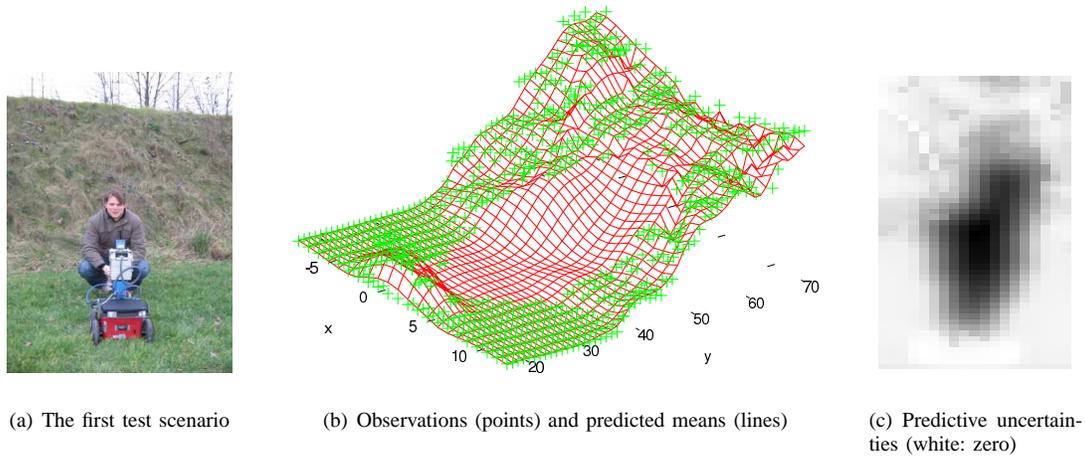


Fig. 5. A real-world scenario, where a person blocks the robot’s view on an inhomogeneous and sloped terrain (a). Figure (b) gives the raw data points as well as the predicted means of our adapted non-stationary regression model. Importantly, our model also yields the predictive uncertainties for the predicted elevations as depicted in Figure (c).

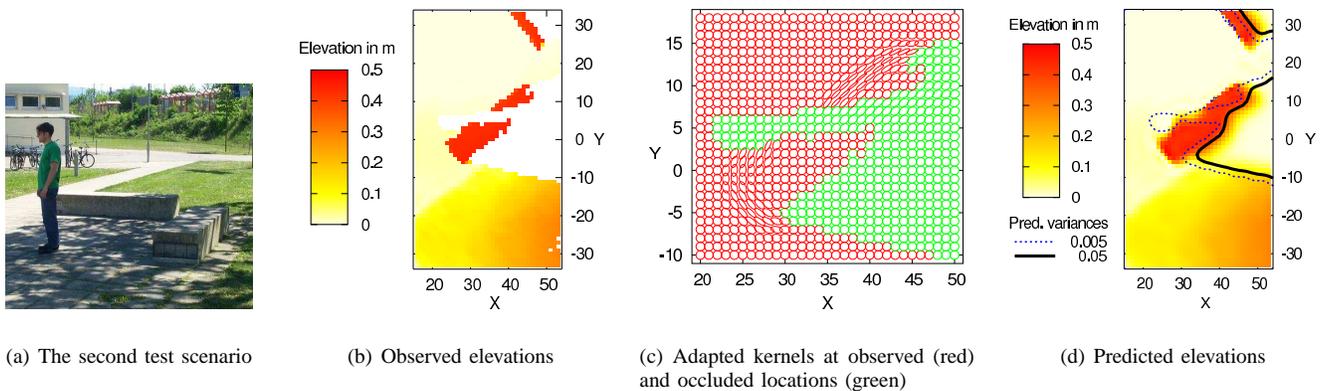


Fig. 6. A real-world scenario where a person blocks the robot’s view on a stone block, i.e., a sharp linear discontinuity (a). Figure (b) visualizes the kernels that have adapted to the observed block edges illustrated in (c). Figure (d) illustrates the predicted terrain elevations and two contour lines for two different predictive uncertainty thresholds.

## REFERENCES

- [1] John Bares, Martial Hebert, Takeo Kanade, Eric Krotkov, Tom Mitchell, Reid Simmons, and William Red L. Whittaker. Ambler: An autonomous rover for planetary exploration. *IEEE Computer*, 22(6):18–26, June 1989.
- [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] D. Cornford, I. Nabney, and C. Williams. Adding constrained discontinuities to gaussian process models of wind fields. In *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, 1999.
- [4] Christian Früh, Siddharth Jain, and Avideh Zakhori. Data processing algorithms for generating textured 3d building facade meshes from laser scans and camera images. *International Journal of Computer Vision*, 61(2):159–184, 2005.
- [5] Marco Hugentobler. *Terrain Modelling with Triangle Based Free-Form Surfaces*. PhD thesis, University of Zurich, 2004.
- [6] M. Middendorf and H. Nagel. Empirically convergent adaptive estimation of grayvalue structure tensors. In *Proceedings of the 24th DAGM Symposium on Pattern Recognition*, pages 66–74, London, UK, 2002. Springer-Verlag.
- [7] C. Paciorek and M. Schervish. Nonstationary covariance functions for Gaussian process regression. In *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, 2004.
- [8] P. Pfaff and W. Burgard. An efficient extension of elevation maps for outdoor terrain mapping. In *Proc. of the Int. Conf. on Field and Service Robotics (FSR)*, pages 165–176, Port Douglas, QLD, Australia, 2005.
- [9] 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.
- [10] C. Rasmussen and Z. Ghahramani. Infinite mixtures of Gaussian process experts. In *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, 2002.
- [11] C. E. Rasmussen and C. K.I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, Massachusetts, 2006.
- [12] Paul D. Sampson and Peter Guttorp. Nonparametric estimation of nonstationary spatial covariance structure. *Journal of the American Statistical Association*, 87(417):108–119, 1992.
- [13] Alexandra M. Schmidt and Anthony O’Hagan. Bayesian inference for nonstationary spatial covariance structure via spatial deformations. *Journal of the Royal Statistical Society, Series B*, 65:745–758, 2003.
- [14] H. Takeda, S. Farsiu, and P. Milanfar. Kernel regression for image processing and reconstruction. *IEEE Trans. on Image Processing*, 2006. To appear.
- [15] V. Tresp. Mixtures of Gaussian processes. In *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, 2000.
- [16] R. Triebel, P. Pfaff, and W. Burgard. Multi-level surface maps for outdoor terrain mapping and loop closing. In *Proc. of the International Conference on Intelligent Robots and Systems (IROS)*, 2006.
- [17] O. Williams. A switched Gaussian process for estimating disparity and segmentation in binocular stereo. In *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, 2006.