

On the Interpolation of Data with Normally Distributed Uncertainty for Visualization

Steven Schlegel, Nico Korn, and Gerik Scheuermann, *Member, IEEE*

Abstract—In many fields of science or engineering, we are confronted with uncertain data. For that reason, the visualization of uncertainty received a lot of attention, especially in recent years. In the majority of cases, Gaussian distributions are used to describe uncertain behavior, because they are able to model many phenomena encountered in science. Therefore, in most applications uncertain data is (or is assumed to be) Gaussian distributed. If such uncertain data is given on fixed positions, the question of interpolation arises for many visualization approaches. In this paper, we analyze the effects of the usual linear interpolation schemes for visualization of Gaussian distributed data. In addition, we demonstrate that methods known in geostatistics and machine learning have favorable properties for visualization purposes in this case.

Index Terms—Gaussian process, uncertainty, interpolation.

1 INTRODUCTION

Almost all data suffer from some kind of uncertainty and it is always important to know whether there is uncertainty and how large it is. Otherwise, we risk making decisions that rely on data we should not rely upon or doubting data that we think might be uncertain when it is not.

This can obviously lead to poor decision making. Therefore, it is vital to integrate our knowledge about the uncertainty of the data into our visualizations. As we will see, this not only helps to raise awareness for the existence and magnitude of the uncertainty, but will also allow us to make better estimates about the data than simply assuming a single dataset to be the truth.

An important open problem in uncertainty visualization and related fields is how uncertainty and interpolation interact. This is particularly relevant for data that suffers from missing values or low resolution. There are many established interpolation schemes available, but how reliable are the interpolated values? We want to address this problem by analyzing the traditional linear interpolation and its impact on Gaussian distributed data. We will see that there are counter-intuitive and unwanted side effects. Therefore, we will incorporate Gaussian process regression to interpolate uncertain data. By doing so, we will show that the interpolation scheme is independent of the data itself – similar to non-random data. Furthermore, we will give an analytical description of the basis functions for the interpolation.

The rest of the paper is structured as follows. In Section 3, we explore the effects of linear interpolation on the uncertainty of the interpolated values. Section 4 gives the mathematical background for Gaussian process regression and Section 5 shows how this can be used to build a complete stochastic model of arbitrary scalar data that takes our knowledge about the uncertainties and correlations in the data into account. We then show that for every Gaussian process an equivalent interpolation scheme exists. Finally, we tested Gaussian process regression and compared it to classic linear interpolation using synthetic and real-world data in Section 6. We will see, that the quality of interpolation in Gaussian processes is superior in many ways when compared to linear interpolation.

- Steven Schlegel is with the University of Leipzig, E-mail: schlegel@informatik.uni-leipzig.de.
- Nico Korn is with the Helmholtz-Centre Dresden-Rossendorf, E-mail: n.korn@hzdr.de.
- Gerik Scheuermann is with the University of Leipzig, E-mail: scheuermann@informatik.uni-leipzig.de.

Manuscript received 31 March 2012; accepted 1 August 2012; posted online 14 October 2012; mailed on 5 October 2012.

For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

2 RELATED WORK

2.1 Modeling Uncertainty

This paper will deal with Gaussian process regression, a statistics technique that has applications in machine learning [20] and is also known in the GIS scene, there called *Kriging* [10] (named after the South African mining engineer Daniel Krige who developed the method). It can be used to model uncertain data, i.e. a set of Gaussian-distributed random variables. However, it has not been used for general purpose visualization yet and there is not much material for heteroscedastic Gaussian processes, i.e. Gaussian processes with different uncertainty levels at the data points.

There does not seem to be other widespread methods for general purpose uncertainty modeling, indeed not even Kriging seems to be known outside the GIS community. But some approaches for special applications do exist.

For very simple cases, it may be sufficient to simply assume identical distributions, perhaps two-dimensional or more, at every data point, for example a Gaussian/normal, log-normal, exponential, Poisson or some other, and just store the parameters (e.g. (co)variance) at every data point. For more complicated situations, other schemes are needed.

Pauly et al. [16] visualize possible surfaces that fit a given point cloud. To do this, they created their own empirical model to estimate the probability of a line going through a particular point and compute that probability for all pixels. That probability is designed in such a way that there is a low angle between the point and a possible line segment, and a small length of that line segment leads to a high probability. All the probabilities with regard to all line segments are merged into a distance-weighted sum which gives the final probability for the surface to pass through that point. A big advantage of that method is, that distinct alternative routes are possible, which is not usually the case as regression normally seeks to find a single mean function.

Pöthkow et al. [18] presented a generalization of isocontours for uncertain fields. They compute the level-crossing probability – the probability for an interval to have a crossing of the given threshold in it. A simple, but convincing method that is easy to understand and compute. They interpolate the expected values and the roots of the central moments in order to interpolate the probability density function. This work was extended to correlated data by Pöthkow et al. [19].

Inspired by [18], Pfaffelmoser et al. [17] developed an algorithm to incrementally compute *isosurface-first-crossing-probability* (IFCP). This probability is visualized by volume rendering. They enhance the visualization with the rendering of the stochastic distance function (SDF-surfaces).

Kniss et al. [9] try to perform classification of medical volume data under uncertainty. They base their transfer function on what they

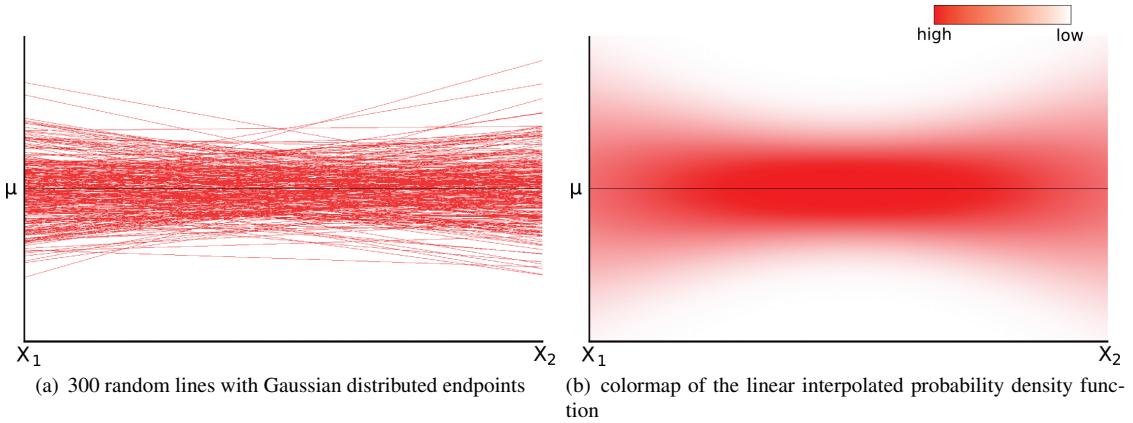


Fig. 1. Illustrations on the interpolation of the probability density function: $X_1 \sim \mathcal{N}(\mu, \sigma^2)$ and $X_2 \sim \mathcal{N}(\mu, \sigma^2)$ are Gaussian distributed random variables.

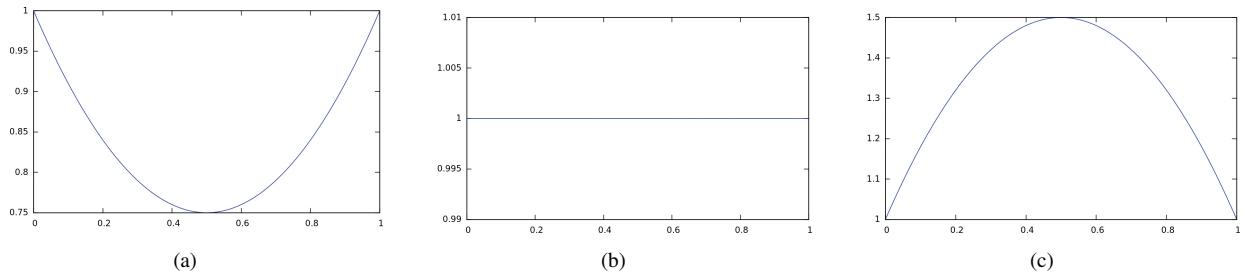


Fig. 2. Interpolation of the variance $\sigma_1^2 = 1$ of X_1 at $s = 0$ and the variance $\sigma_2^2 = 1$ of X_2 at $s = 1$ with (a) $\text{cov}(X_1, X_2) = 0.5$, (b) $\text{cov}(X_1, X_2) = 1$ and (c) $\text{cov}(X_1, X_2) = 2$.

call the decision boundary distance that is computed for every class, which is a maximal log-odds ratio of all the other classes. Roughly speaking, it is a measurement of the risk of being wrong to assume that the current class is the correct one.

2.2 Ambiguity for Uncertainty Visualization

The idea behind intentionally ambiguous visualization is that if the data is uncertain, the viewer should be equally uncertain about the visualized value. There are several ways to achieve this goal, but it is worth noting that they can often be interpreted as visualizations of the probability density function of the uncertain value.

A common method for ambiguous visualization is to make lines or other glyphs “fat”, i.e. scale their width proportionally to the uncertainty. From a probabilistic point of view, this corresponds to a uniform distribution of values (in contrast to a Gaussian blurring). Fat glyphs can be applied to extend a wide array of existing methods to include uncertainty. Li et al. [11] for example, use an ellipse to show the anisotropic uncertainty of the star positions they are visualizing with their system. Lines in 3D become ribbons or tubes, see [23] and [12] for example. More elaborate applications are also possible. Hlawatsch et al. [7] designed a glyph for vector data that shows both the variation of flow over time and the angular uncertainty over time. Since we want to show the results of our approach on scalar data, the following approach will be suitable: For heightfields, fat surfaces can be implemented as multiple surfaces or direct volume renderings. Their inherent problem is that the surfaces will obscure each other and are easily confused. To avoid this, the mean surface can be of different color and high specularity ([18]). Zehner et al. on the other hand [26] added extra geometry to isosurfaces in order to depict positional uncertainty.

Another way of visualizing uncertainty information is to simply not display anything at all or make uncertain data transparent. An interesting application of non-visualization comes from map-making. Lowell et al. [13] only draw boundaries of areas when they are certain about

where they actually are. If they are uncertain, the line dividing two areas simply stops. Strothotte et.al.[22] developed a visualization tool for ancient architecture that avoids photorealism, instead mimicking a line drawing with the strokes becoming thinner, transparent and shaky in areas of high uncertainty. For direct volume rendering, transparency is the natural way to implement non-visualization, see Djurcicov et al. [5] for example.

3 ANALYSIS OF LINEAR, BILINEAR AND TRILINEAR INTERPOLATION OF UNCERTAIN DATA

Most visualization methods for grid-based scalar, vector, or tensor data require interpolation. This is also true if the data comes with Gaussian distributed noise. In this section, we analyze the effect of linear, bilinear, or trilinear interpolation of such data.

First, we consider the 1-d case, i.e. linear interpolation on the line. In the simplest case, we are given two uncorrelated scalar Gaussian distributions $X_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $X_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$ at the positions s_1 and s_2 . This is also the typical case in one dimension, because each line segment is interpolated without any dependence on data outside the line segment. Before we turn to the theory, let us do a simple experiment. We draw a finite number of random lines between these two points, i.e. looking at the linear interpolation between the endpoints. Each line has its endpoints normally distributed according to X_1 and X_2 . It is obvious that more lines will appear that have both of their endpoints near the means of X_1 and X_2 than for instance lines with both endpoints near $2 \cdot \sigma_1$ and $2 \cdot \sigma_2$, respectively. Since we are interpolating, let us have a look at an arbitrary position s in between s_1 and s_2 . Fig. 1(a) shows 300 random lines with Gaussian distributed endpoints. Judging from the image it appears that the variance at s ($s_1 < s < s_2$) is smaller than at the endpoints. In fact, if we consider the value at s as a scalar random variable X , the usual linear interpolation formula $X = X_1\alpha + X_2(1 - \alpha)$ (with interpolation parameter $\alpha \in [0, 1]$) allows

to compute the variance σ^2 of X as

$$\begin{aligned}\sigma^2 &= \alpha^2 \sigma_1^2 + (1-\alpha)^2 \sigma_2^2 + 2(1-\alpha)\alpha \cdot \text{cov}(X_1, X_2) \\ &= \alpha^2 \sigma_1^2 + (1-\alpha)^2 \sigma_2^2\end{aligned}\quad (1)$$

because we assumed that X_1 and X_2 are independent which means $\text{cov}(X_1, X_2) = 0$.

Furthermore, it can be shown (by evaluating the integral over the joint probability density function) that the linear interpolation of two arbitrary, uncorrelated, Gaussian-distributed variables $X_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $X_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$ yields another Gaussian-distributed random variable X with

$$X \sim \mathcal{N}(\mu_1 \alpha + \mu_2 (1-\alpha), \sigma_1^2 \alpha^2 + \sigma_2^2 (1-\alpha)^2) \quad (2)$$

We see that the mean of X is acquired by linear interpolation of μ_1 and μ_2 . For the variance, we can observe that it is the result of quadratic interpolation of σ_1^2 and σ_2^2 . This interpolation function has its local maxima at the points where X_1 and X_2 are given, i.e. the gridpoints in visualization terms. Consequently, the minimum variance lies somewhere in between the two points depending on σ_1^2 and σ_2^2 . An illustrative example for the interpolation of the probability density function is shown in 1(b).

This theoretical result appears counter-intuitive, but it is a consequence of the linear interpolation of independent, Gaussian-distributed variables: the (weighted) average of two random variables is always more stable than each of those variables (in the sense of having less variance). We leave it as an easy task to the reader to extend this result to the linear interpolation of three Gaussian variables in a triangle, or four Gaussian variables in a tetrahedron.

So far, we have only considered uncorrelated Gaussian variables. Examining Eq. 1 yields that the covariance influences the behavior of the variance of the interpolated values. We can see that, depending on the values of σ_1^2 , σ_2^2 and $\text{cov}(X_1, X_2)$, the interpolation function behaves differently (see Fig. 2):

- If $\sigma_1^2 + \sigma_2^2 > 2 \cdot \text{cov}(X_1, X_2)$, we have the behavior described above (e.g. see Fig. 2(a)).
- If $\sigma_1^2 + \sigma_2^2 = 2 \cdot \text{cov}(X_1, X_2)$, the quadratic interpolation of the variance reduces to linear interpolation (e.g. see Fig. 2(b)).
- If $\sigma_1^2 + \sigma_2^2 < 2 \cdot \text{cov}(X_1, X_2)$, the described behavior is negated, i.e. the variance has its maximum between X_1 and X_2 and its local minima at X_1 and X_2 (e.g. see Fig. 2(c)).

Those findings can be expanded to the bilinear case in two dimensions, see Fig. 3 for example.

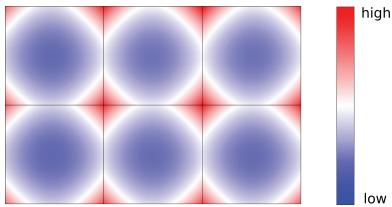


Fig. 3. color coded variance as the result of linear interpolation of uncorrelated Gaussian distributed variables given on the grid points.

We want to point out that this problem is not restricted to linear interpolation. Interpolation with cubic splines for example also suffer from those implications. Consider a Bezier curve given by

$$Y = X_1(1-\alpha)^3 + X_2 3\alpha(1-\alpha)^2 + X_3 3\alpha^2(1-\alpha) + X_4 \alpha^3. \quad (3)$$

Assuming that X_1, X_2, X_3, X_4 are independent, the variance along this curve is given by

$$\begin{aligned}\text{var}(Y) &= \text{var}(X_1) * (1-\alpha)^6 + \text{var}(X_2) * 9\alpha^2(1-\alpha)^4 \\ &\quad + \text{var}(X_3) * 9\alpha^4(1-\alpha)^2 + \text{var}(X_4) * \alpha^6.\end{aligned}\quad (4)$$

Plotting this function (for $0 \leq \alpha \leq 1$), would reveal similar results: depending of the value of the variances of X_1, \dots, X_4 , the interpolated variance has its (local) maxima at the positions of the given data points.

By our belief, traditional interpolation (regardless of the specific interpolation scheme) is in general not well suited to estimate uncertain values. A key element of interpolation is that the interpolation result equals the value at the data points. In the case that the value is uncertain, this property may be unwanted because the variance of that data point is disregarded. And the result of the variance might be that the best estimator for that data point is actually not the given value at that data point.

The goal of this section was to point out that the traditional linear interpolation, which is widely used in grid based visualization, may yield very counter-intuitive and unwanted results when dealing with uncertain data. If the covariance between neighbouring points is weaker than the variance at these points, then the resulting interpolated data is most certain between the grid points (where we have no data) and the variance has its maxima at the grid points (where the data is given). If one encounters such data, it should be obvious by now that one needs other methods. Such methods are presented in the following sections.

4 GAUSSIAN PROCESSES

As denoted earlier, we suggest to use an interpolation scheme based on suitable mathematical foundations without the described deficits of linear interpolation. It turns out that stochastic processes provide such a foundation. Because we focus on Gaussian-distributed data, we will concentrate on Gaussian processes.

4.1 Interpolation Task

The visualization of normally-distributed scalar, vector, and tensor field data typically faces the following setting: There is a domain S which is a nice subset (stratified manifold for the mathematically inclined reader) of \mathbb{R}^2 or \mathbb{R}^3 . The data is given at a finite number of positions $s_1, \dots, s_N \in S$. At these positions, we are given normally-distributed random variables X_1, \dots, X_N as data. We denote by μ_1, \dots, μ_N the means and by $\sigma_1^2, \dots, \sigma_N^2$ the variances at each position. Basically, interpolation means to define normally distributed random variables at all points $s \in S$. In addition, properties of the interpolation may be demanded, like smoothness, limited variation between the positions, or minimal interpolation error. Posed differently, we are looking for a distribution of fields over the domain S where the values at each position $s \in S$ have a Gaussian distribution that is in some (yet to define) sense related to the given distributions at the positions s_1, \dots, s_N . Gaussian processes are exactly the mathematical concept to provide all this [2, 20]. In the following, we assume that we are given a distribution of scalar values and are looking for a scalar Gaussian process.

4.2 Gaussian Processes

Actually, our linear interpolation of normally distributed values in the last section was already a Gaussian process.

Definition: Multivariate Gaussian

A random variable $X = (X_1, \dots, X_n) \in \mathbb{R}^d$ is **multivariate Gaussian** if any linear combination of X_i is univariate Gaussian, i.e. $a^T X = \sum_{i=1}^n a_i X_i \sim \mathcal{N}(\mu, \sigma^2)$, $\forall a \in \mathbb{R}^n$ and suitable $\mu, \sigma \in (\mathbb{R})$.

Definition: Gaussian Process

For any domain S , a **Gaussian Process (GP)** f on S is a set of random variables $(f_t : t \in S)$, such that $\forall n \in \mathbb{N}$ and $\forall t_1, \dots, t_n \in S$, $(f_{t_1}, \dots, f_{t_n})$ is a multivariate Gaussian distributed random variable.

At first glance, this definition is directed more towards pure mathematics, but there is practical use to the concept. A first hint is given by the fact that one just needs to define a mean value at each point $s \in S$ and a covariance between any two points $s, s' \in S$ to define a unique Gaussian process, (see [2, p.5]). If we have a nice set S , the positions

s_1, \dots, s_N and scalar Gaussian distributed data X_1, \dots, X_N as before, we need a mean function and covariance function, i.e.

$$\begin{aligned} \mu : S \rightarrow \mathbb{R} \quad \mu(s) &= \mathbb{E}[f(s)], \\ k : S \times S \rightarrow \mathbb{R} \quad k(s, s') &= \mathbb{E}[(f(s) - \mu(s))(f(s') - \mu(s'))], \end{aligned} \quad (5)$$

to uniquely define a scalar Gaussian process f on S . For the distributions of the value at an arbitrary position $s \in S$, we have

$$f(s) \sim \mathcal{N}(\mu(s), k(s, s)). \quad (6)$$

because the variance at any position is the covariance with itself.

4.3 Interpolation Scheme Defines Covariance Function

In this and the next subsection, we want to show that choosing an interpolation scheme like the linear, bilinear or trilinear interpolations in Section 3 is equivalent to choosing a covariance function. This equivalence is the key to understand the considerations in this article!

Let S, s_1, \dots, s_N , and X_1, \dots, X_N be our data as before. Any interpolation scheme in visualization can be described the Lagrangian way by a set of weight functions $\phi_1, \dots, \phi_N : S \rightarrow \mathbb{R}$ with the defining properties

$$\phi_i(s_j) = \delta_{ij} \quad \forall i, j = 1, \dots, N \quad (7)$$

$$\sum_{i=1}^N \phi_i(s) = 1 \quad \forall s \in S \quad (8)$$

where δ_{ij} denotes the Kronecker delta. We define the Gaussian process by (see [21])

$$\begin{aligned} f : S &\rightarrow \mathbb{R} \\ s &\mapsto \sum_{i=1}^N X_i \phi_i(s). \end{aligned} \quad (9)$$

As mentioned above, a Gaussian process is uniquely defined by its mean function and its covariance function. For the mean function $\mu : S \rightarrow \mathbb{R}$, we get

$$\begin{aligned} \mathbb{E}[f(s)] &= \mathbb{E}\left[\sum_{i=1}^N X_i \phi_i(s)\right] \\ &= \sum_{i=1}^N \mathbb{E}[X_i] \phi_i(s) \\ &= \sum_{i=1}^N \mu_i \phi_i(s) := \mu(s) \end{aligned} \quad (10)$$

Obviously, the mean of the data is interpolated by the weight functions. For the covariance function $k : S \times S \rightarrow \mathbb{R}$, we obtain

$$\begin{aligned} \text{cov}(f(s), f(s')) &= \mathbb{E}[(f(s) - \mu(s))(f(s') - \mu(s'))] \\ &= \mathbb{E}\left[\left(\sum_{i=1}^N (X_i - \mu_i) \phi_i(s)\right) \left(\sum_{j=1}^N (X_j - \mu_j) \phi_j(s')\right)\right] \\ &= \sum_{i,j=1}^N \mathbb{E}[(X_i - \mu_i)(X_j - \mu_j)] \phi_i(s) \phi_j(s') \\ &= \sum_{i,j=1}^N k_{ij} \phi_i(s) \phi_j(s'). \end{aligned} \quad (11)$$

where k_{ij} is the covariance between the data distributions X_i and X_j . You can see that the covariances are interpolated by products of weight functions. This leads to the quadratic behavior of the variance of linear interpolation schemes which creates the strange effects noted in the previous section.

4.4 Covariance Function Defines Interpolation Scheme

In the previous subsection, we showed that interpolation schemes in visualization are actually Gaussian processes if applied to normally distributed data. Basically, the Lagrangian weight functions of the

scheme determine the covariance function of the Gaussian process. The interesting fact is that one can also perform this construction in the opposite direction ([1, pp.17-19]). This means that we will start with a covariance function and derive a corresponding interpolation scheme. Of course, one can also use such a construction for approximation by assuming some (zero mean Gaussian) error of the data. We will do this in the next section and it should be no surprise as we assume that we have uncertainty in the data. From a visualization point of view, this might look useless at first, because it is not clear where the covariance function should come from. However, using covariance functions is actually the method of choice in other disciplines like geospatial statistics or machine learning. In geospatial statistics, the method is called Kriging. A classical presentation of this approach was published by Cressie [4]. In machine learning, the method is called Gaussian process regression [20].

Let us assume that we have a bilinear, symmetric, and positive definite function $k : S \times S \rightarrow \mathbb{R}$. k is our covariance function. We want to define a set of basis functions that allow us to calculate weights from our data to derive a Gaussian process with this covariance function. For this purpose, we define the set of functions

$$H = \{g : S \rightarrow \mathbb{R} | g(s) = \sum_{i=1}^M a_i k(s, t_i), M \in \mathbb{N}, t_i \in S, a_i \in \mathbb{R}\} \quad (12)$$

It should be noted that we really allow any $M \in \mathbb{N}$ and any points $t_i \in S$ in this space. On this set, we define the scalar product

$$\begin{aligned} \langle f, g \rangle &= \left\langle \sum_{i=1}^N a_i k(t, t_i), \sum_{j=1}^N b_j k(t, t_j) \right\rangle \\ &= \sum_{i=1}^N \sum_{j=1}^N a_i b_j k(t_i, t_j) \end{aligned} \quad (13)$$

which has the so called *reproducing kernel property*

$$\langle f(\cdot), k(t, \cdot) \rangle = f(t) \quad (14)$$

This construction is also called the *reproducing kernel map* construction (see [20]). Furthermore, we extend H to its completion under its scalar product which is the so-called **reproducing kernel Hilbert space (RKHS)**. The Moore-Aronszain Theorem ([3]) promises a one-to-one correspondence between k and the RKHS. We can now take any basis

$$\{\phi_i : S \rightarrow \mathbb{R}, \quad i = 1, \dots, \dim(H)\} \quad (15)$$

to define our approximation scheme. However, for many covariance functions, $\dim(H)$ will be infinite, so we define as possible Gaussian processes

$$g : S \rightarrow \mathbb{R}, \quad g(s) = \sum_{i \in I} A_i \phi_i(s) \quad (16)$$

for a finite subset $I \subset \{1, \dots, \dim(H)\}$ and Gaussian-distributed random variables A_i . Every g will have k as its covariance function, so we can set the ϕ_i to fit our data according to some approximation condition. This will be shown in the next section.

5 GAUSSIAN PROCESS REGRESSION

5.1 Regression in Gaussian Processes

Now we show how we can use the results of Section 4 to give an analytical description for the interpolation of the Gaussian-distributed variables $X_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ at the gridpoints s_i . The continuous uncertain tensorfield will be conceived as a Gaussian process. Information on the distribution of the X_i can be derived empirical, for example if we have a high number of realizations of a random experiment or a high number of simulations. In some cases we may even be able to derive the Gaussian distributions directly.

We assume the standard linear model for the data:

$$f(s) = y(s) + \varepsilon(s), \quad (17)$$

where $f(s)$ is the observed value at an arbitrary position s , $y(s)$ is the true (but unknown) value at s and $\varepsilon(s)$ is the Gaussian-distributed error with zero mean and variance $\sigma^2(s)$. We assume that the error and the true value are uncorrelated.

5.1.1 Covariance Function

In Section 4, we learned that the key for the interpolation is the choice of a covariance function $k(s, s')$. Unfortunately an exact formulation for the covariance of the Gaussian process exists only in very rare cases. Therefore, the covariance is often modeled by a specific covariance function. Which function fits best for the data varies from case to case. If not denoted otherwise, we will use the squared exponential as our covariance function throughout this paper:

$$k(s, s') = \exp\left(-\frac{1}{2}|s - s'|^2\right). \quad (18)$$

It is interesting to see, that by using the squared exponential, the covariance does not depend on the data itself but only on the positions. We can even compute a covariance between two arbitrary points whether we have data about them or not. The use of the squared exponential is very common throughout the Gaussian process literature, see [14, p.14] for explanation. Pfaffelmoser et al. [17] also chose an exponential function for modeling correlation. As we see, the covariance function models *spatial* correlation between the data points. It turns out that this is the case with most covariance functions [20]. In many applications the covariance function is adjusted with extra parameters, that permit more flexibility when modeling the data. These parameters are called hyperparameters. The squared exponential, for instance, can have the following form

$$k(s, s') = \sigma_p^2 \exp\left(-\frac{1}{2l^2}|s - s'|^2\right), \quad (\sigma_p^2, l > 0) \quad (19)$$

The parameters are σ_p^2 – the signal variance and l – the length scale. We see that the overall variance can be controlled by σ_p^2 as it is a positive prefactor. The length scale determines the width of the squared exponential, i.e. how far away one point has to be from another point, so that their influence on each other is negligible. As the squared exponential has the shape of a Gaussian bell curve, l is similar to the variance in a Gaussian density function. Therefore, the squared exponential is also referred to as Gaussian kernel.

To derive suitable hyperparameters is a topic on its own. The ideal case would be, that we can derive the hyperparameters directly from a given model or theory (just like the covariance function). But this is often not the case. So the hyperparameters have to be estimated from the data. A natural choice for the parameter σ_p^2 is the maximum variance of our input data. A common method to evaluate the length scale is to try to maximize the log-marginal-likelihood of the Gaussian process for various values of l (see [20, p.123]). Roughly speaking, we use different values for l and compute the likelihood of the data to actually show up with that particular value for l . At last we take the value for l that produced the highest likelihood.

When doing regression in Gaussian processes, it is important to specify a *prior* distribution. The prior gives us information at points in the field, where we have no data. If we do not have specific information about the prior, we can use heuristics. The prior mean is taken to be zero. This is the so-called simple Kriging assumption. We can model a zero mean process on our data by subtracting the empirical mean from the input data and add it back after processing, if necessary. For Kriging, it is important to have a constant mean throughout the input domain. An obvious (and widely used) heuristic for the prior variance is the maximum empirical variance found in the dataset. It is loosely speaking a worst case approximation of the variance at a position without any data. And the worst case which we can anticipate, given the data, is the maximum variance in the data. Now we can see that the parameter σ_p^2 in the squared exponential is in fact the prior variance, because $k(s, s) = \sigma_p^2$ for arbitrary s in our domain.

5.1.2 The Covariance Matrix

Now that we have a covariance function, we can compute the covariance matrix K :

$$K = \begin{pmatrix} k(s_1, s_1) & \dots & k(s_1, s_N) \\ \dots & \dots & \dots \\ k(s_N, s_1) & \dots & k(s_N, s_N) \end{pmatrix}, \quad (20)$$

where $k(s_i, s_j)$ ($i, j = 1, \dots, N$) is the covariance between the i -th and j -th datapoint and can be evaluated by using the covariance function. It is important to note that for every diagonal entry, the variance has to be added to the covariance. So we get

$$K_{ij} = \begin{cases} k(s_i, s_j) + \sigma_i^2, & i = j \\ k(s_i, s_j), & i \neq j \end{cases}. \quad (21)$$

That is because (see eq. 17)

$$\begin{aligned} \text{Cov}(f(s), f(s')) &= \text{Cov}(y(s) + \varepsilon(s), y(s') + \varepsilon(s')) \\ &= \text{Cov}(y(s), y(s')) + \text{Cov}(y(s), \varepsilon(s')) \\ &\quad + \text{Cov}(\varepsilon(s), y(s')) + \text{Cov}(\varepsilon(s), \varepsilon(s')) \end{aligned} \quad (22)$$

The error ε and the true value y are uncorrelated, so we get

$$\begin{aligned} \text{Cov}(f(s), f(s')) &= \text{Cov}(y(s), y(s')) + 0 + 0 + \text{Cov}(\varepsilon(s), \varepsilon(s')) \\ &= k(s, s') + \text{Cov}(\varepsilon(s), \varepsilon(s')) \\ &= \begin{cases} k(s, s') + \sigma^2(s), & s = s' \\ k(s, s'), & s \neq s' \end{cases} \end{aligned} \quad (23)$$

In the next section, it can be noticed that adding the variances to the diagonal of the covariance matrix in fact incorporates the uncertainty in the interpolation scheme and - loosely speaking - turns the interpolation into a regression.

We want to mention that it is possible not to construct the covariance using the covariance function but using the empirical covariances. Despite the implications on the model that is assumed for the uncertain data it might be the case that this matrix cannot be inverted. In the next section, we will see that we need to invert the covariance matrix in order to compute the interpolation.

5.1.3 Gaussian Process Regression

The interpolation of the observed values will be defined as (see eq. 16)

$$f(s) = \sum_{i=1}^N f_i \phi_i(s), \quad (24)$$

where f_i are the observed values at the datapoints s_i .

When using Gaussian Process regression, the weight functions $\phi_i(s)$ will be chosen in a way that the estimated value $f(s)$ minimizes the variance of the interpolation error. Before we can minimize that variance, we will show how it can be computed. $X(s)$ is the real Gaussian-distributed value at s and $X^*(s)$ denotes the result of the interpolation. The error is therefore given by $X^*(s) - X(s)$ and its variance by:

$$\begin{aligned} \sigma^2(s) &= \text{var}(X^*(s) - X(s)) \\ &= \mathbb{E}[(X^*(s) - X(s))^2] - \mathbb{E}[X^*(s) - X(s)]^2 \end{aligned} \quad (25)$$

Since we assume a zero mean Gaussian process, we get

$$\mathbb{E}[X^*(s) - X(s)] = \mathbb{E}[X^*(s)] - \mathbb{E}[X(s)] = 0 \quad (26)$$

Therefore, we get for the variance:

$$\begin{aligned} \sigma^2(s) &= \mathbb{E}[(X^*(s) - X(s))^2] \\ &= \mathbb{E}[X^*(s)X^*(s)] - 2\mathbb{E}[X^*(s)X(s)] + \mathbb{E}[X(s)X(s)] \end{aligned} \quad (27)$$

We can evaluate this formula, by using the fact that $\mathbb{E}[X(s)X(s')] = \text{Cov}(X(s), X(s')) = k(s, s')$ (because of the zero mean Gaussian process) and by using equation 11:

$$\sigma^2(s) = \sum_{i,j=1}^N \phi_i(s)\phi_j(s)k(s_i, s_j) - 2 \sum_{i=1}^N \phi_i(s)k(s_i, s) + k(s, s) \quad (28)$$

To minimize the variance, we have to take the derivative of eq. 28 with respect to the weight functions and set it to zero:

$$\begin{aligned} & \sum_{j=1}^N \phi_j(s)k(s_i, s_j) - k(s_i, s) = 0, \quad \forall i = 1 \dots N \\ \Leftrightarrow & \sum_{j=1}^N \phi_j(s)k(s_i, s_j) = k(s_i, s), \quad \forall i = 1 \dots N \end{aligned} \quad (29)$$

This equation induces a linear system (involving the covariance matrix K), which can be solved to compute the optimal weights $\phi_i(s)$:

$$\begin{pmatrix} k(s_1, s_1) & \dots & k(s_1, s_N) \\ \dots & \dots & \dots \\ k(s_N, s_1) & \dots & k(s_N, s_N) \end{pmatrix} * \begin{pmatrix} \phi_1(s) \\ \dots \\ \phi_N(s) \end{pmatrix} = \begin{pmatrix} k(s_1, s) \\ \dots \\ k(s_N, s) \end{pmatrix} \quad (30)$$

Now that we have the optimal weight functions, we can compute $f(s)$ (see eq. 24). The variance can be computed by inserting eq. 29 into eq. 28:

$$\sigma^2(s) = k(s, s) - \sum_{i=1}^N \sum_{j=1}^N \phi_i(s)\phi_j(s)k(s_i, s_j) = k(s, s) - \sum_{i=1}^N \phi_i(s)k(s_i, s), \quad (31)$$

which can be evaluated, if the optimal weights were computed using eq. 30. We want to remind that $k(s, s)$ is the prior variance, which is σ_p^2 if using the squared exponential.

Evaluating this formula is very inefficient, because for every s , a linear system of size $N \times N$ has to be solved. For that reason it is common to invert the covariance matrix once (it does not change for varying s) and compute the basis functions as

$$\phi_i(s) = \sum_{j=1}^N (K_{ij})^{-1} k(s_j, s), \quad (32)$$

where $(K_{ij})^{-1}$ are the elements of the inverted covariance matrix.

To summarize, the outcome of the Gaussian process at s is a Gaussian-distributed variable

$$X^*(s) \sim \mathcal{N} \left(\sum_{i=1}^N \phi_i(s)f_i, \quad k(s, s) - \sum_{i=1}^N \phi_i(s)k(s_i, s) \right). \quad (33)$$

The basis functions $\phi_i(s)$ are computed as in eq. 32.

As a side note, we want to mention that eq. 31 is the so-called Kriging variance. There is a discussion in the Kriging community regarding the interpretation of the kriging variance (e.g. see [15, 25] and references therein) and also papers discussing alternatives for the variance of the interpolated Gaussian process (e.g. see [24]).

5.1.4 Model Flexibility

A big concern on the regression in Gaussian processes might be that we need to have a constant mean and one covariance function throughout the input domain. But these restrictions can be overcome by considering the uncertain field not as one Gaussian process but as many Gaussian processes with smaller input domains (they also may overlap). To accomplish this, we only have to figure out which of the neighboring data points s_i are to be considered for the calculation of the distribution at s .

For example, let the squared exponential be the covariance function. Depending on the parameter l , positions that have a large Euclidian

distance have a covariance that is almost zero. So they have a negligible influence on the interpolation result. That means these points can be left out. Since the length scale l is similar to the variance in a Gaussian density function, the data points within a $3l$ neighborhood contribute more than 99% to the interpolation result. So we can leave out data points lying outside of that range without losing too much precision. We see that it may be sufficient to consider only $n < N$ data points in order to compute the interpolation result.

There are two main advantages using this approach. Firstly, we can assign each Gaussian process its own mean and its own covariance function. That enables us to work with random fields where the spatial correlation cannot be described by one covariance function. This also applies, if we have to use different hyperparameters for the squared exponential, for instance. Secondly, there is an advantage regarding the computational costs. For every random field, we have to invert the covariance matrix, which is of size $N \times N$. Using a decomposition in many smaller processes leads to the inversion of many smaller covariance matrices. That in general is much faster to compute and moreover can be done in parallel. Furthermore, we can assume that within a grid cell, the Gaussian process does not change significantly. That means, that the data points that contribute (significantly) to the interpolation result do not vary within the cell. Thus we can compute one covariance matrix for each cell (and not for each interpolation point within the cell). So for each cell, we can precompute an inverted covariance matrix that enables us to compute the interpolation result very fast compared to the inversion of a $N \times N$ matrix. Especially multi-core computers will benefit from this approach, since each cell can be treated in parallel.

Regarding the varying length scale, we want to note that Gibbs [6] proposed an exponential covariance function, where the length scale is in fact a function of the position s . That shows it is also possible to vary the length scale within the uncertain field by using a suitable covariance function.

5.2 Example

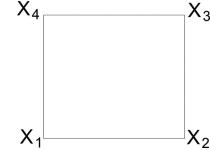


Fig. 4. Interpolation in a quad cell.

Consider the quad cell given in Fig. 4. The distributions on the grid points are the Gaussian distributed values X_1, X_2, X_3, X_4 with $f_1 = 1, f_2 = 1, f_3 = -1, f_4 = 0$ and $\sigma_i^2 = 1, i = 1, \dots, 4$. The quad cell is a square with a side length of 1. This is important to know, if we want to adjust our covariance function. Recall, that the covariance function used (the squared exponential k) is a function of the distance. The mean of the GP is zero. The hyperparameter σ_p^2 will be the maximum variance, which is 1. As a start, we will set the other parameter l to 0.2, which indicates a very short covariance function and results in very small covariance between the 4 given points. To admit that the observations at the grid points are uncertain, we have to add the variance to the covariance matrix, i.e. $\text{cov}(s_i, s_j) = k(s_i, s_j) + \sigma_i^2 \delta_{ij}$.

In Fig. 5(a), we can see the result of the interpolation. There are two important observations. At first, we can see that the values at s_i are not the observed values f_i , because they are assumed to be an uncertain observation. Loosely speaking, they tend towards the prior mean, which is zero. And the “degree of tendency” is determined by the variance. That means the higher the variance is, the more the f_i tend towards zero. We can conclude that the best estimator for $f(s_i)$ is usually not f_i . Another observation is, that the samples tend towards the prior mean in the middle of the cell. This is due to the short covariance function. The given data points barely influence each other, so that they tend towards the prior mean very fast. The variance tends

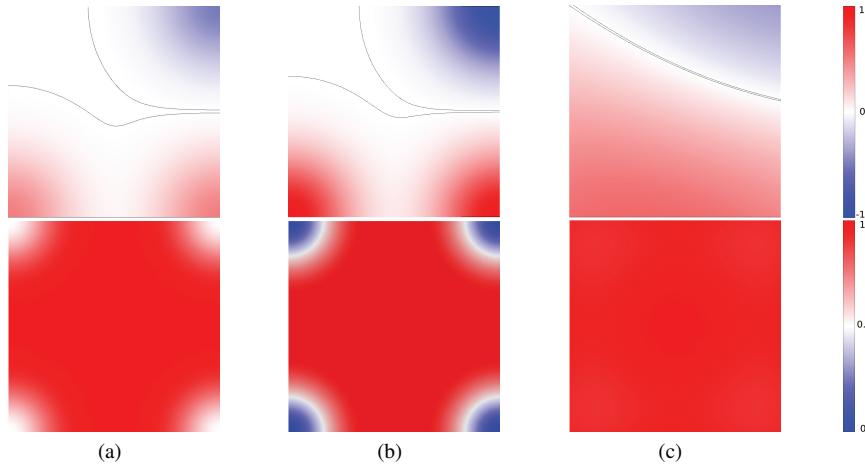


Fig. 5. Interpolation in a sample quad cell using squared exponential kernel with (a) $l = 0.2$ and $\sigma_p^2 = 1$, (b) $l = 0.2$ and $\sigma_p^2 = 1$ but certain input ($\sigma_i^2 = 0, i = 1, \dots, 4$), and (c) $l = 0.7$ and $\sigma_p^2 = 1$. The upper row depicts interpolated values $f(s)$, the lower row depicts the corresponding variance $\sigma^2(s)$. The black isolines in the upper pictures enclose the area that is numerically evaluated to 0.

towards the prior variance accordingly. In Fig. 5(b), we set the variance of the inputs to zero. This implies, that the 4 observations f_i of the Gaussian process are certain. But to admit that the interpolated values are uncertain, we set the prior variance to 1. This is of course a rather theoretical assumption to have certain input and the rest of the field is uncertain. But it serves the purpose to show that the interpolation result behaves as we would expect it. Comparing with Fig. 5(a), we can see that the values at the data points are not changed anymore, because they are assumed to be certain. In this case the best estimator for $f(s_i)$ is indeed f_i . The grid points in Fig. 5(c) have uncertain input values, like in Fig. 5(a). But the length scale l was changed to 0.7. For the samples, we can observe that the area which is evaluated to zero became thinner. That is because the given data at the data points gain more influence on the interpolated values within the cell, because of the broader covariance function. As the values at the data points f_i are not all equal to zero, so are the interpolated values within the cell. For the corresponding variance, we can see that it is almost constant within the cell.

6 RESULTS

6.1 Comparing Gaussian Process Regression and Linear Interpolation

6.1.1 Synthetic Dataset

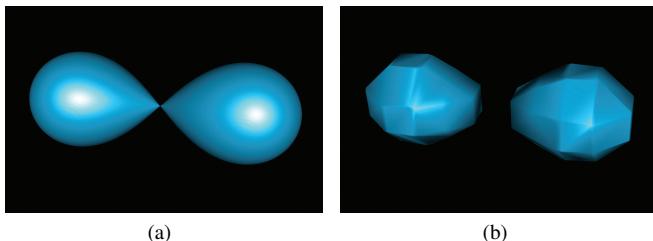


Fig. 6. Isosurfaces (isovalue=1) of the synthetic scalar field used for evaluation and comparison. 6(a) shows the original 3D scalar field. 6(b) shows a much lower resolution version of the same field, which we used as a test case.

We tested Gaussian process regression against trilinear interpolation using a simple synthetic 3D scalar field f defined by

$$f(s) = |p - s| |q - s| \quad (34)$$

where $p = (-1, 0, 0)$, $q = (1, 0, 0)$. The value of this field is zero near the points p and q and increases with distance. The isosurface for

the value $f = 1$ (shown in Fig. 6(a)) is particularly interesting, because at this value two separate isosurfaces surrounding p and q respectively connect at $(0, 0, 0)$. This is a difficult, but important situation. To put the interpolation capabilities of Gaussian process regression to the test, we applied it to a very-low-resolution ($12 \times 12 \times 12$) version of the field as shown in Fig. 6(b). Adding to the difficulty of the task, the critical point at $(0, 0, 0)$ lies in the center between 4 grid points, so the low-resolution field contains no information about this point. The variance of the data was defined to be 0.1 on the whole field. For the Gaussian process, the prior variance was set to be 0.2 and for the length scale, a value of 0.3 was chosen, which corresponds to the cell width of the grid.

Fig. 7(b) shows the surface where the posterior mean produced by the Gaussian process, i.e. the expected value equals one. It very closely resembles the original field (Fig. 6(a)). It is smooth but at the same time reconstructed the critical point at $(0, 0, 0)$ pretty well, even though the low-resolution field did not even include a sample at that position. The trilinear interpolation (Fig. 7(e)) on the other hand fails to reconstruct the field near the critical point properly and is much less smooth.

The Gaussian process regression gives us a mean and a variance for every point in the field. We chose to compute and visualize the probability for the field to fall below the interesting value 1: $P(f < 1)$. The results can be seen in Fig. 7 for different probabilities. Enclosed by the isosurfaces are the areas where the probability $P(f < 1)$ is greater than the given thresholds 95%, 50% and 5% (from left to right). The results based on Gaussian processes (top row of Fig. 7) are consistently closer to reality than those based on trilinear interpolation (bottom row). In particular, the latter suffers from heavy artifacts as described in Section 3.

6.1.2 Climate Dataset

We also want to show the results of the interpolation on simulations of the Intergovernmental Panel of Climate Change carried out at DKRZ by the Max-Planck-Institute for Meteorology. In particular, we have used the IPCC A1B scenario results [8]. We used the 2m temperature dataset.

The data is placed on a uniform 192×96 grid. The side length of a cell is 1. Each gridpoint has a time series of the simulated monthly means of the years 1860-2100. We constructed a Gaussian distribution on each gridpoint by eliminating the annual cycle and removing the linear trend of each time series. We checked the result with the Kolmogorow-Smirnov-Test. The prior mean was set to the empirical mean of all of these time series and for the prior variance, we chose the maximum variance in the dataset. The covariance function is again the squared

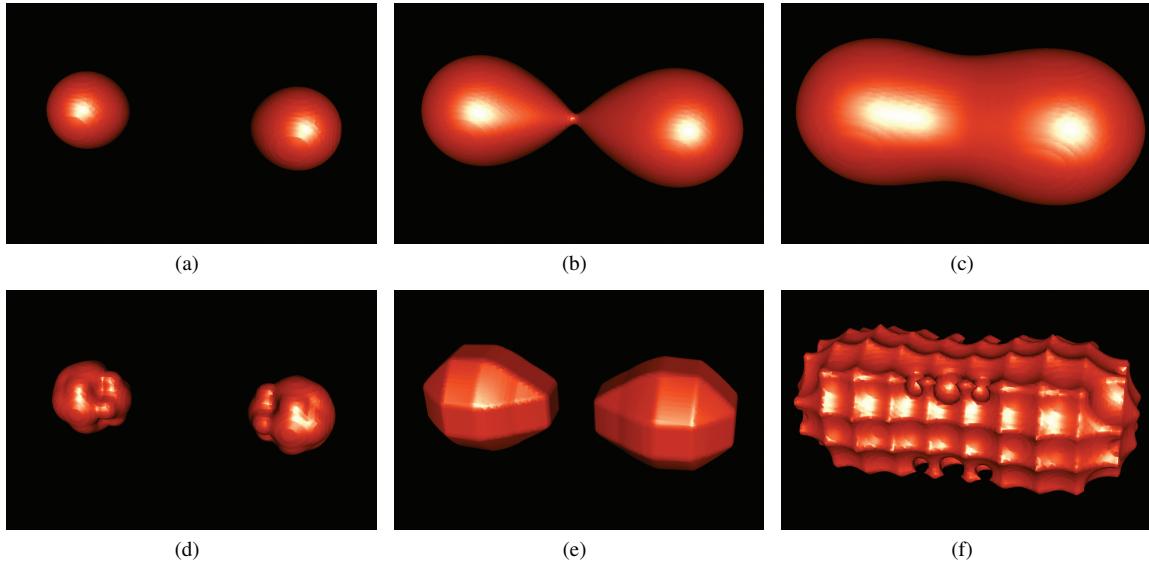


Fig. 7. Results of Gaussian Process regression (top row) compared to trilinear interpolation (bottom row). The images show the probability that the value of the field is below 1, i.e. $P(f < 1)$ for different isovalues: (a) and (d) $P(f < 1) = 95\%$, (b) and (e) $P(f < 1) = 50\%$, (c) and (f) $P(f < 1) = 5\%$. For example, the balls in 7(a) contain all the points that have more than 95% probability to have a value smaller than 1.

exponential. We interpolate the empirical means and the variances at the gridpoints for a part of the northern american continent. The results are shown in Fig. 8. Fig. 8(a) depicts the linear interpolation of the means. The according interpolation result for the variance is displayed in Fig. 8(d). The other images show the regression of the Gaussian process with differing values for l . We can see, that the variance of the linear interpolation (Fig. 8(d)) is highest at the grid points. In Fig. 8(e), we have a short covariance function ($l = 0.7$) and the variance is highest within the cells. Also, smoothing effects can be seen when using a longer covariance function (Fig. 8(c) and 8(f)).

6.2 The Level-Crossing Probability

The level-crossing probability (LCP) field was introduced by Pöthkow et al. ([18], [19]) and shows the probability that a certain level γ is crossed between two sample points. It is used to depict the positional uncertainty of an isocontour. By visualizing this field using volume rendering, one can identify regions that have a higher or lower probability to be intersected by the uncertain isocontour.

We used the low-sampled ($12 \times 12 \times 12$) synthetic data set from the previous subsection (with the same setup) and calculated the LCP field for $\gamma = 1$ based on mean and variance values from Gaussian process regression. We used three different values for the length scale to investigate its influence on the LCP. Fig. 9 shows direct volume renderings of the LCP fields and the corresponding isosurface, i.e. the surface where the mean equals $\gamma = 1$.

We succeeded in recovering the smooth underlying structure of the original data. It can be seen, that the LCP values become smaller as the length scale increases. This is because of the increased covariance between the points, which decreases the variances and thus the positional uncertainty of the isocontour. This behavior is consistent with the findings of [19].

7 SUMMARY AND CONCLUSIONS

We showed that the traditional linear interpolation yields counter-intuitive results when dealing with uncertain data. To overcome these problems, we consider the uncertain tensorfield to be a Gaussian process which is defined by a covariance function. That allows to apply methods widely used especially in geostatistics and machine learning. We showed how to derive an interpolation scheme using Gaussian process regression to describe the continuous uncertain field. This enables us to interpolate Gaussian-distributed variables. We showed that the

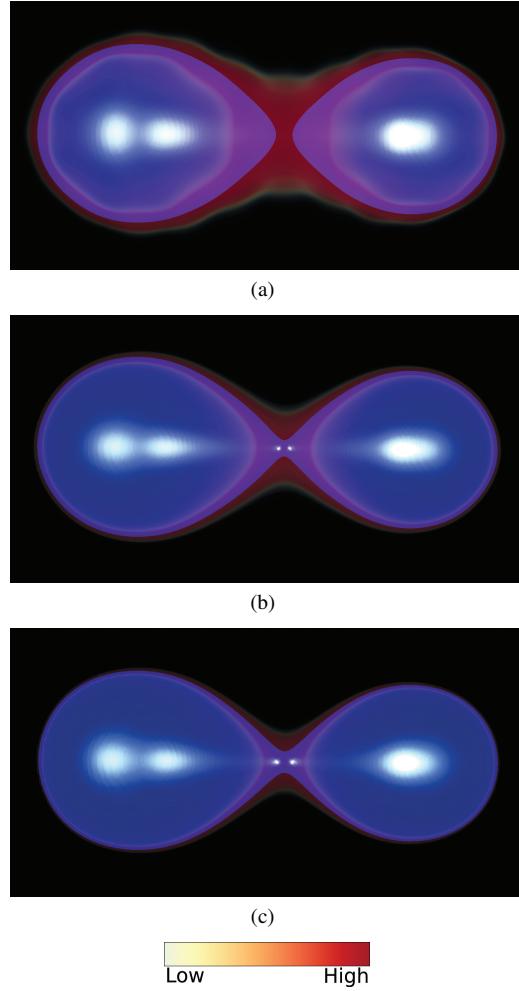


Fig. 9. LCP fields using squared exponential as covariance function: (a) with $l = 0.3$ (b) with $l = 0.6$, and (c) with $l = 0.9$. LCP is depicted by direct volume rendering, additionally the mean surface for $\gamma = 1$ is shown in blue (transparent).

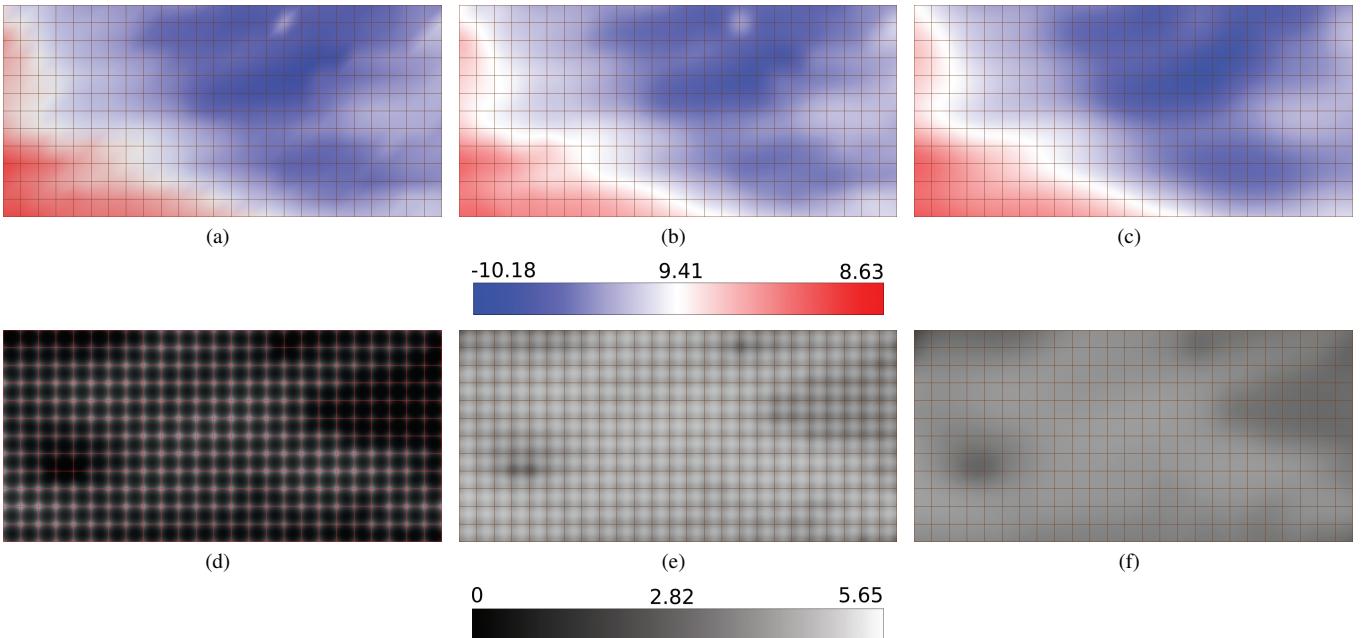


Fig. 8. Interpolation of a climate dataset: (a) and (d) linear interpolation with $l = 0.7$, (b) and (e) Gaussian process regression with $l = 0.7$, (c) and (f) Gaussian process regression with $l = 1.5$. The upper row depicts the means of the GP $\mu(s)$, the lower row depicts the corresponding variances $\sigma^2(s)$.

implications from linear interpolation do not arise anymore, especially if the Gaussian variables are weakly correlated.

An important property of the Gaussian process regression is, that the basis functions of the interpolation are independent of the data itself – as it is the case for the standard interpolation schemes. They depend on the chosen covariance function. So choosing the covariance function is similar to choosing an interpolation scheme. For the non-random case this equals the question, if for example one should use linear interpolation or B-splines on the data. And by providing the analytical description of the basis functions, we are able to implement Gaussian process regression independently of the given tensor field (e.g. for the use in visualization tools). These conclusions and the description of the basis functions were – to our knowledge – missing in the Kriging literature.

In contrast to traditional interpolation methods, Gaussian processes do not merely create an isostructure that runs through the data points, but respects the uncertainty in them. This way, noise, errors or outliers in the data do not disturb the model inappropriately. Most importantly, the model shows the variance in the interpolated values, which can be higher but also lower than that of its neighbouring data points. That provides us with a lot more insight into the quality of our data and how it influences our uncertainty. In Section 6.1, we showed that Gaussian process regression is able reconstruct a low sampled grid and also to find a critical point within a grid cell. Furthermore, Gaussian processes work without a grid. The data can be a point cloud without any topology or neighbouring information whatsoever as long as there is a meaningful covariance function.

The biggest concerns (especially in the visualization community) about Gaussian process regression might be its complexity and its choice of parameters, i.e the covariance function. Nevertheless, we believe that the GIS community and the machine learning community provide us with reasonable heuristics in order to find suitable covariance functions: MacKay and Gibbs for example [14, 6] give good reasons why the covariance structure in a Gaussian process can be estimated by an exponential function. Heuristics and algorithms for the estimation of hyperparameters are mentioned in Section 5.1.1. The GIS community often uses the variogram to fit a correlation function to the data.

A more detailed research on the free parameters – especially for ap-

plications in visualization – is beyond the scope of this article and will be part of our future work. One of the problems, that also are to address in the future, is to extend the general interpolation scheme to arbitrary stochastic processes in order to apply the interpolation scheme to other models that exhibit non-gaussian noise.

ACKNOWLEDGMENTS

We want to thank Michael Böttger who is with the DKRZ for providing the climate data set.

REFERENCES

- [1] R. Adler and J. Taylor. *Topological complexity of smooth random functions: École d’Été de Probabilités de Saint-Flour XXXIX - 2009*. Lecture Notes in Mathematics. Springer, 2011.
- [2] R. J. Adler and J. E. Taylor. *Random Fields and Geometry (Springer Monographs in Mathematics)*. Springer, June 2007.
- [3] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68, 1950.
- [4] N. Cressie. The origins of kriging. *Mathematical Geology*, 22(3):239–252, Apr. 1990.
- [5] S. Djuricov, K. Kim, P. Lermusiaux, and A. Pang. Visualizing scalar volumetric data with uncertainty. *Computers & Graphics*, 26(2):239–248, Apr. 2002.
- [6] M. N. Gibbs and D. J. C. MacKay. Efficient implementation of gaussian processes. submitted to Statistics and Computing.
- [7] M. Hlawatsch, P. Leube, W. Nowak, and D. Weiskopf. Flow radar glyphs; static visualization of unsteady flow with uncertainty. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):1949–1958, 2011.
- [8] Intergovernmental. *Climate Change 2007 - The Physical Science Basis: Working Group I Contribution to the Fourth Assessment Report of the IPCC*. Cambridge University Press, Cambridge, UK and New York, NY, USA, Sept. 2007.
- [9] J. M. Kniss, R. Van Uitert, A. Stephens, G. S. Li, T. Tasdizen, and C. Hansen. Statistically quantitative volume visualization. pages 287–294, Oct. 2005.
- [10] D. G. Krige. A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Chemical Metallurgical and Mining Society of South Africa*, 52(6):119–139, 1951.
- [11] H. Li, C.-W. Fu, Y. Li, and A. J. Hanson. Visualizing large-scale uncertainty in astrophysical data. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1640–1647, Nov. 2007.

- [12] A. Lopes and K. Brodlie. Accuracy in 3d particle tracing. In *In Hans-Christian Hege and Konrad Poltheir, editors, Mathematical Visualization*, pages 329–341, 1998.
- [13] K. E. Lowell and P. Casault. Outside-in, inside-out: Two methods of generating spatial certainty maps. In *University of Otago*, pages 26–29, 1997.
- [14] D. J. C. MacKay. Introduction to Gaussian processes. In C. M. Bishop, editor, *Neural Networks and Machine Learning*, NATO ASI Series, pages 133–166. Kluwer, 1998.
- [15] M. Monteiro da Rocha and J. K. Yamamoto. Comparison between kriging variance and interpolation variance as uncertainty measurements in the capanema iron mine, state of minas geraisbrazil. *Natural Resources Research*, 9:223–235, 2000. 10.1023/A:1010195701968.
- [16] M. Pauly, N. Mitra, and L. Guibas. Uncertainty and variability in point cloud surface data, 2004.
- [17] T. Pfaffelmoser, M. Reitinger, and R. Westermann. Visualizing the positional and geometrical variability of isosurfaces in uncertain scalar fields. In *Computer Graphics Forum*, volume 30, pages 951–960. Wiley Online Library, 2011.
- [18] K. Pöthkow and H.-C. Hege. Positional uncertainty of isocontours: Condition analysis and probabilistic measures. *IEEE Transactions on Visualization and Computer Graphics*, 17(10):1393 – 1406, 2011.
- [19] K. Pöthkow, B. Weber, and H.-C. Hege. Probabilistic marching cubes. *Computer Graphics Forum*, 30(3):931 – 940, 2011.
- [20] C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- [21] G. Scheuermann, M. Hlawitschka, C. Garth, and H. Hagen. Mathematical foundations of uncertain field visualization. submitted to Dagstuhl Follow Ups, Workshop on Scientific Visualization, Seminar 11231, 2011.
- [22] T. Strothotte, M. Masuch, and T. Isenberg. Visualizing knowledge about virtual reconstructions of ancient architecture. *Computer Graphics International Conference*, 0:36–43, 1999.
- [23] C. Wittenbrink, E. Saxon, J. J. Furman, A. Pang, and S. Lodha. Glyphs for visualizing uncertainty in environmental vector fields. In *IEEE Transactions on Visualization and Computer Graphics*, volume 2410, pages 266–279, 1995.
- [24] J. Yamamoto. An alternative measure of the reliability of ordinary kriging estimates. *Mathematical Geology*, 32:489–509, 2000. 10.1023/A:1007577916868.
- [25] J. Yin, S. H. Ng, and K. M. Ng. A study on the effects of parameter estimation on kriging model's prediction error in stochastic simulations. In *Winter Simulation Conference, WSC '09*, pages 674–685. Winter Simulation Conference, 2009.
- [26] B. Zehner, N. Watanabe, and O. Kolditz. Visualization of gridded scalar data with uncertainty in geosciences. *Computers and Geosciences*, 36(10):1268–1275, 2010.