Measuring Event Probabilities in Uncertain Scalar Datasets using Gaussian Processes

Steven Schlegel
University of Leipzig
Augustusplatz 10
04109 Leipzig, Germany
schlegel@informatik.uni-leipzig.de

Sebastian Volke
University of Leipzig
Augustusplatz 10
04109 Leipzig, Germany
volke@informatik.uni-leipzig.de

Gerik Scheuermann
University of Leipzig
Augustusplatz 10
04109 Leipzig, Germany
schleuermann@informatik.uni-leipzig.de

ABSTRACT

In this paper, we show how the concept of Gaussian process regression can be used to determine potential events in scalar data sets. As a showcase, we will investigate climate data sets in order to identify potential extrem weather events by deriving the probabilities of their appearances. The method is implemented directly on the GPU to ensure interactive frame rates and pixel precise visualizations. We will see, that this approach is especially well suited for sparse sampled data because of its reconstruction properties.

Keywords

Gaussian Process Regression, OpenCL Programming, Climate Data

1 INTRODUCTION

The visualization of tensor data that is given on discrete positions typically requires interpolation of the data values in between the sample positions. Usually, this task is solved by using linear interpolation. However, in the case that the given data is uncertain, this method is not feasible [4]. In [21], Schlegel et al. proposed the use of Gaussian process regression to overcome the aforementioned problems.

We will show how this method can be used to calculate occurence probabilities of certain events. Therefore, we present an approach that combines fast computation on the GPU as well as visualizations of arbitrarily dense samplings. This is achieved by using the reconstruction properties of Gaussian Process Regression. We use datasets from the climate research domain to demonstrate our method. One of the major tasks in climate research is the prediction and the understanding of extreme weather events. Events like heat waves, heavy precipitation (resulting in floods) or hurricanes have a large impact on society and politics. Decision makers rely on climate simulation results as accurate as possible. Those results should hold characteristics of extreme events like location, frequency and intensity. There is a lot of research that points out that climate and extreme events undergo a change especially in frequency and intensity. Emanuel [5], for instance, pointed out that the destructiveness of hurricanes immensely increased of the past 30 years. A quick overview for other observed changes can be viewed here: `http://www.ipcc.ch/publications_and_data/ar4/wg2/en/ch10s10-2-3.html#table-10-3`.

2 RELATED WORK

As pointed out by [13] it is important to keep in mind that uncertainty of scalar data can reside in the data value or in the position of the data point or in both. In this paper, we deal with the uncertainty of the data value itself. Pöthkow et al. [16] also targeted this issue and presented an uncertain counterpart for isocontours [12]. Therefore, they calculate the so called level-crossing probability (LCP). In a given interval, the probability is computed that a certain threshold is crossed within. Therefore, they interpolated the expected values and the roots of the central moments to interpolate the probability density function. Schlegel et al. [21] proposed the method of Kriging to interpolate the mean and the variance in an uncertain Gaussian field. They also applied their method to compute LCP. Several acceleration methods were employed in [23] to enable a fast computation of Kriging in 3D scalar fields. They created interactive 3D visualizations of the mean field and showed the confidence of the computation by depicting areas of high uncertainty. Therefore, they computed an upper boundary for the posterior variance. In contrast, we aim to provide probabilities for the data to exceed (or fall below) certain thresholds using the exact posterior variance. [2] analyzed the effects of uncertainty to linear interpolation and isosurface extraction. The extension of [16] to correlated data was done in [19]. To reduce the heavy computation time (mainly caused by Monte Carlo Sampling), two methods called maximum edge crossing probability and linked-pairs to approximate the level-crossing probabilities were introduced in [18]. To overcome the restrictions of predefined probability distributions [17] introduced nonpara-

metric models (empirical distributions, histograms, and kernel density estimators) to compute the probabilty of features in an uncertain field. Based on [16], Pfaffelmoser et al. [15] developed an algorithm to compute the so called isosurface first crossing probability. It is an algorithm that incrementally uses a front-to-back volume ray casting to visualize that probability. The rendering is enriched by additionally depicting surfaces of the stochastic distance function (SDF-surfaces). Additional work to compute the gradient of the probability density function of uncertain 3D scalar fields was done in [14]. Kniss et al. [9] try to perform classification of medical volume data under uncertainty. They base their transfer function on what they 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. A more thorough overwiev of visualization of uncertain data can be read in [3].

3 GAUSSIAN PROCESS REGRESSION

As pointed out earlier, we can not rely on standard techniques like linear interpolation, when the need for interpolation of uncertain data arises, . We need to regard the uncertainty of the sampled data points, as well as their respective corellation. If these samples are Gaussian-distributed random variables, it is suitable to use the concept of *Gaussian processes*. Interpolating random variables in a Gaussian process is also known as *Kriging* [10] or *Gaussian process regression* [20]. The basic approach is to assume a prior Gaussian distribution for any (continuous) position in the data set. By considering the given samples and a defined covariance between the data points, this prior distribution is turned into a posterior Gaussian distribution that matches the given data more precisely in the sense of reducing the variance of the distribution. For details, on how to define the prior distribution, we refer to [21].

A Gaussian process given on a domain S defines a Gaussian distributed random variable at any position $s \in S$. It is defined by a mean function at every position s and a covariance function between any two positions s and s', e.g., see [1]:

$$\begin{aligned} \mu &: S \mapsto \mathbb{R} \qquad \mu(s) = \mathbb{E}[f(s)], \\ k &: S \times S \to \mathbb{R} \quad k(s,s') = \mathbb{E}[(f(s)-\mu(s))(f(s')-\mu(s'))], \end{aligned} \tag{1}$$

where the mean function is assumed to be constant. Our choice for the covariance function throughout this paper is the squared exponential. This covariance function models an exponential drop of the covariance with increasing distance of the data points. It is often used in the field of Geostatistics and is given by

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

The parameters σ_p^2 (prior variance) and l (length scale) are hyperparameters. Throughout this paper our choice for l will be 1. The choice for σ_p^2 will be discussed in section 4.1. Gaussian processes, as well as the optimization of the hyperparameters, are discussed in detail in [20].

Let S be sampled with N Gaussian distributed variables at positions $s_i,\ i = 1,\ldots,N$, with $X(s_i) = X_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right)$ and the covariance function $k(s,s')$. Then one can calculate the covariances between those sample points and generate the covariance matrix

$$K = \begin{pmatrix} k(s_1,s_1)+\sigma_1^2 & \ldots & k(s_1,s_N) \\ \ldots & \ldots & \ldots \\ k(s_N,s_1) & \ldots & k(s_N,s_N)+\sigma_n^2 \end{pmatrix}, \tag{3}$$

The posterior distribution at position s is then defined as

$$X(s) \sim \mathcal{N}\left(\vec{k(s)}^T K^{-1}\vec{\mu}_i, \quad k(s,s) - \vec{k(s)}^T K^{-1}\vec{k(s)}\right), \tag{4}$$

with $\vec{\mu}_i$ being the vector of the means of the sampled X_i and $\vec{k(s)} = (k(s,s_1),\ldots,k(s,s_N))^T$. It can be shown that the variance of the posterior distribution is minimized, when estimating $X(s)$ in that way.

Additionaly, by defining the basis functions (see [23])

$$\phi_i(s) = \begin{cases} -1, & \text{if } i = 0 \\ \sum_{j=1}^N (K^{-1})_{ij} k(s_j,s), & \text{otherwise} \end{cases} \tag{5}$$

we can write the computation of $X(s)$ as

$$X(s) = \sum_{i=0}^{N} X_i \phi_i(s). \tag{6}$$

with

$$\begin{aligned} X(s) \sim \mathcal{N}\Big(\mu(s) &= \sum_{i=1}^{N} \phi_i(s)\mu_i, \\ \sigma^2(s) &= k(s,s) - \sum_{i=1}^{N} \phi_i(s) k(s_i,s)\Big). \end{aligned} \tag{7}$$

Using Eq. 6, it is possible to compute the derivative of $X(s)$ by differentiating the basis functions:

$$\begin{aligned} \frac{\delta\mu(s)}{\delta s^{(n)}} &= \mathbb{E}\left(\sum_{i=1}^{N} \frac{\delta\phi_i(s)}{\delta s^{(n)}} X_i\right), \\ \frac{\delta\sigma^2(s)}{\delta s^{(n)}} &= Var\left(\sum_{i=1}^{N} \frac{\delta\phi_i(s)}{\delta s^{(n)}} X_i\right) \end{aligned} \tag{8}$$

This form of Kriging is called *simple* Kriging. Other forms of Kriging differ mainly in the form of the assumption of the prior (e.g. see [6]).

4 METHOD

4.1 Modeling the Gaussian Process

In this section, we show how to work with Gaussian regression on climate research datasets. Our goal is to calculate the probability that a certain threshold will (not) be exceeded. Thus, we construct uncertainty variables on the basis of the original climate data. We use time dependent data from a global climate simulation given on a rectilinear 2D grid. This could be interpreted as a time series of data at each grid position.

First we normalize each time series by removing the seasonal component. Normalized time series have a Gaussian distribution [11]. So the normalization enables us to compute the variability of the data and, of course, to apply our method. Therefore, we replace the value at each time step $v(ts)$ with the average of these values from the annual cycle. For example, if we have monthly means, we replace $v(ts)$ with the average of the period $[ts-6;ts+6]$. Furthermore, we can remove a linear trend in the dataset by replacing each time step with its forward difference, i.e. $v(ts) = v(ts+1) - v(ts)$. Although the trend in temperature time series is not necessarily linear, we can use this simplification for relatively short time periods (i.e. 30 years). This step is optional and best suited for datasets where linear trends may disturb the normalization (e.g. temperature data). In order to estimate the variability of the simulated data at that position, we derive the empirical variance for every grid point based on the normalized time series.

As described in section 3, the basic principle of Gaussian process regression is to turn a prior Gaussian distribution, which we observe on the data, into a posterior Gaussian distribution by taking the given samples and the covariance into account. The prior distribution describes the uncertainty in the data acquisition method. The posterior distribution on the other hand is in general a better estimator for the uncertainty in the dataset in the sense of having less variance. To model the prior distribution, we need it's mean and it's variance. The mean is constantly zero. This can be accomplished by subtracting the empirical mean of the data from the samples. When we display the results, we simply add the posterior mean back on each sample point. This applies to an error model, where the observed value is the sum of the true (unknown) value and a zero mean Gaussian error. The prior distribution variance is the maximum of all the variances which we extracted at the grid points. The maximum variance in the dataset is an obvious choice for the prior variance, because the variance of the data acquisition method is at least as big as the maximum variance residing in the dataset. The prior variance (or signal variance) is the factor σ_p^2 in the covariance function, see eq. 2, which results in $k(s,s) = \sigma_p^2$ (see eq. 4).

Algorithm 1 Creating The Cell Cache on the CPU

```
l := length scale
d := cell diameter
n := number of Cells
CellCache[n]
for i = 0 to n do
    b := barycenter(Cell)
    P := all sample points in radius [b − (3l + d), b + (3l + d)]
    create and invert covariance matrix using all points in P
    CellCache[i] := inverted covariance matrix, its positions, and its samples
end for
send CellCache to graphics card
for all pixels do
    calculateColor()
end for
```

Algorithm 2 "calculateColor()" – GPU Colormap Algorithm.

```
t := threshold
idx := cell index of Pixel
pos := position in (2D) world space of Pixel
n := number of sample points in CellCache[idx]
for i = 0 to n do
    compute basisfunction φ_i using p and CellCache[idx]
end for
compute distribution using the basisfunctions and Eq. 7
calculate probability p that the value at pos falls below t
color pixel according to p and given color map
```

4.2 Implementation

Gaussian process regression performs poorly on many datasets. The reason is the storage and the inversion of the covariance matrix. A method to reduce those requirements, is the use of many small Gaussian processes (and thus covariance matrices) instead of one large process. For regular sampled datasets, it is feasible to create a small Gaussian process for each grid cell composed of the data points lying in a $3l+d$ radius of the bary center of the cell. Where l is the length scale of the covariance function and d is the diameter of the cell. This approach is described in more detail in [21]. The result is, that we have to invert one relatively small covariance matrix for every cell instead of one large matrix, which can also be done in parallel for another speed up.

When an inverted covariance matrix (see Eq. 3) for each cell of the dataset is computed, we send those matrices to the GPU. We also store the dataset itself as well as the indices of the data points that belong to each of those local Gaussian processes on the GPU. The next step is to compute the probability distribution according to eq. 4 for every pixel position which lies inside our dataset domain. Now, we are able to compute probabilities that values at that pixel fall below or exceed certain thresholds (which in our case are indices for extreme weather events). Furthermore, we are able to accumulate those probabilities over several time steps in order to compute probabilities that the values fall below or exceed the threshold over a given period of time. Given the fact, that we compute everything on the GPU, we finally use the given probabilities for each pixel to create a pixel precise color map which

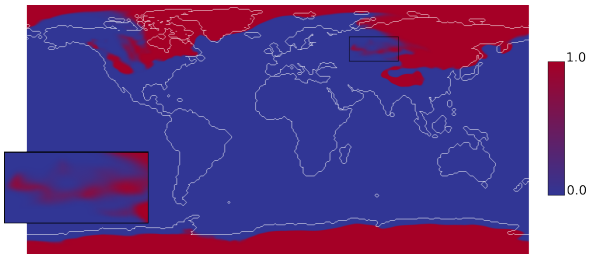


Figure 1: Colormap of the probability that the temperature of 273.15°K (0°C) is not exceeded in the whole month of January 2001.

can be rendered immediately by writing the color into the frame buffer. The main advantage of this approach is, that we send the required data (inverted covariance matrix and the point indices) to the GPU once, which will process both tasks, namely computing and rendering. There is no need to send the data back to the CPU. Our implementation uses the OpenCL framework. The algorithm's pseudocode for processing the data on the CPU to send it to the GPU is given in algorithm 1. The pseudocode for the GPU implementation of the calculateColor() function is depicted in algorithm 2.

5 RESULTS

The data we use is a temperature data set from a global climate simulation of IPCC scenario A1B with the coupled atmosphere ocean general circulation model (AOGCM) ECHAM5-MPIOM, which was carried out as a contribution to the International Panel on Climate Change Assessment Report 4 (IPCC AR4) [8]. It is given on a 192 x 96 rectilinear grid. At each grid point, there is a temperature data time series of monthly means from the year 1860 to the year 2100. We used the data from 1860 to 1890 in order to calculate the variances for every grid position.

After the variances are calculated, we assigned them to temperature data (simulated by the same model) given on a 6 hour basis to calculate the probability that the temperature of 273.15°K (0°C) was not exceeded in the whole month of January 2001 (124 time steps). The prior distribution is calculated as described in section 4.1. The result is given in Fig. 1. Additionally, we zoomed into one area containing probability transitions to demonstrate that this kind of interpolation in fact enables rendering using arbitrary zoom factors and still providing smooth results.

This kind of application is also interesting with respect to regional climate changes. Therefore, we used as a second example a data set from a simulation with the regional climate model CLM [7]. This is a community model for the German climate research, originally based on the LM forecast model of the German Weather Service (DWD). The CLM simulation was

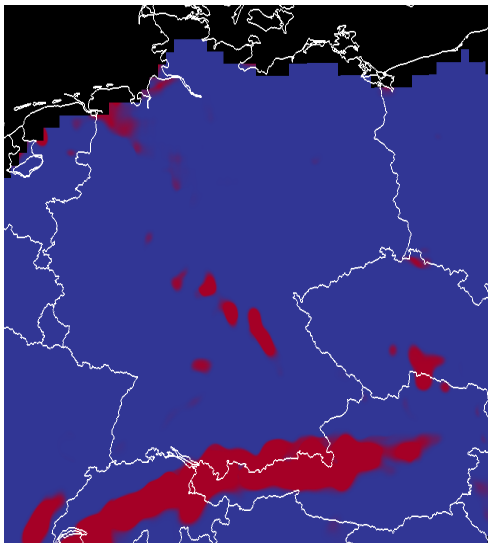
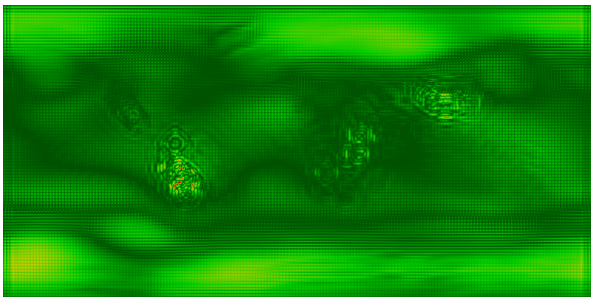


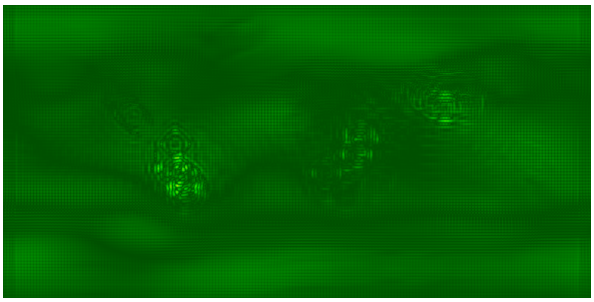
Figure 2: Colormap of the probability that the surface runoff exceeds a threshold of 60 kg/m^2 at least five consecutive days in the summer of 1961.

forced with results of the IPCC scenario A1B simulation with ECHAM5 / MPI-OM. The particular dataset we used is the surface runoff. The surface runoff is the amount of water that cannot be absorbed by the soil. It is an accumulated quantity mainly based on precipitation, snow melting, and the water content of the soil surface and is an indicator for floods. If large volumes of surface runoff flow into a river in a short period of time, the likeliness of a flooding increases. In climate research, one typically counts how often the data exceeds or falls below a threshold within a certain interval to identify weather extremes; see [24] and references. As in the example above, we can normalize the time series data at each grid point and calculate the probability that a certain threshold of the surface runoff is exceeded. With our method, we are able to interpolate the data, incorporate the uncertainty of the simulations into the interpolation and compute the probability pixel by pixel. A second step would be to assign the results to the corresponding river catchment basin and accumulate the probabilities over this area to derive potential risks for people living near those rivers, see [22]. Unfortunately this is beyond the scope of this paper.

The data we used is a cutout of 65×50 grid points of daily CLM data for Europe centered on Germany. The grid is regular (data stream 3) and has a spacing of 0.2° (approximately 20km). It is a simulation run for the 20th century (20C) from 1961-1990. In Fig. 2, we depicted the probability that the surface runoff exceeds the amount of 60 kg/m^2 in at least five consecutive days in the summer (June, July, and August) of 1961 as a showcase. We can judge from the image that within Germany especially regions inside the catchment basin of the river Rhine have a high probability of exceed-



(a)



(b)

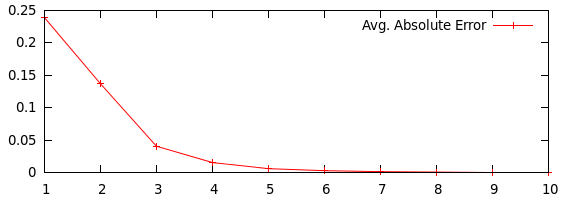
Figure 3: Images showing the regression error for $k = 2$ (Fig.3(a)) and $k = 3$ (Fig.3(b)).

ing the threshold. This method can be a valuable tool when performing research on larger time scales to evaluate the development of such quantities in order to draw conclusions on climate change.

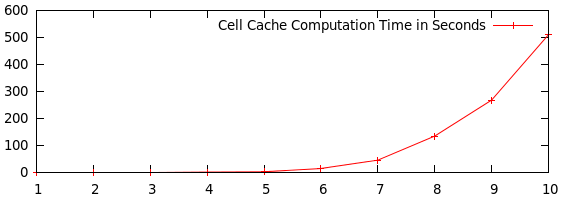
5.1 Error Analysis

In Section 4.2, we showed that cutting off the exponential covariance function provides smaller cell caches and thus a faster computation of the regression result. On the other hand, this technique introduces errors, which we will analyze using a sample 2D climate data set (sea level pressure) on a 192x96 grid. Therefore, we first calculate the inverted covariance matrix for all the grid points, i.e. we do not cut off the covariance function. Then, we do an regression of the samples with this covariance matrix and use this as a ground truth. The interpolated field again is regulary sampled at 2880x1440 positions. In the next step, the field is interpolated and resampled the same way but using shortened covariance functions, each with a different length: $kl + d,\ k = 1,..,10$. Finally, we compare those fields with the ground truth and calculate the absolute average error.

An error colormap for $k = 2$ and $k = 3$ can be seen in Fig. 3. We can conclude from those images, that the error in fact decreases, when the covariance function gains a larger influence radius. But this comes at a comparably high computational cost. The development of error and computational cost to create the cell cache is depicted in Fig. 4. We can conclude an exponential



(a)



(b)

Figure 4: Diagrams showing the exponential drop of the error with increasing covariance influence radius (Fig.4(a)), as well as the exponential gain of computation time (Fig. 4(b)) for different k.

drop of the error as well as an exponential increasing computation time with increasing k. The average relative error for this particular field ranges from $0,0057\%$ to $4,27\%$.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we showed how climate data can be interpolated in an arbitrarily dense matter. Therefore, we use the framework presented in [21] as a basis and extend it by implementing it on the GPU. This enables interactive visualizations with arbitrarily dense samplings. We normalized the time series data to extract the simulation variance. Then we were able to calculate probabilities for certain occurrences like the appearance of extreme events while considering the variance. The resulting color maps can be computed in any desired resolution. Especially when the underlying data is sparse (like in Fig. 2), we nevertheless are able to provide visualizations with smooth transitions.

We want to point out that the variance we used in this paper is computed from the given data. Of course there are cases, when the variance may known a priori, for example when the simulation model has a known error. The proposed method also works with that kind of uncertainty as long as it is Gaussian distributed.

In general, Gaussian process regression works on any type of scalar data regardless of the underlying topological structure. The only precondition is, that there has to be covariance defined between any of the data points in the given domain. This covariance is often modeled with a covariance function, which then serves as the interpolation kernel. Since we use Gaussian process regression for the means of data interpolation, it

is suitable to use a distance based covariance function. With these prerequisites, Gaussian process regression resembles inverse distance based interpolation methods (e.g. Shepard interpolation).

A suitable application of this paper is to study the probabilities of extreme weather events. As mentioned before in Sec. 5, this work can therefore be extended by assigning the calculated probabilities to certain areas of interest to draw conclusions on the danger for flooding, droughts et cetera. We consider this as future work.

7 REFERENCES

[1] R.J. Adler and J.E. Taylor. *Topological complexity of smooth random functions: École d'Été de Probabilités de Saint-Flour XXXIX - 2009*. Lecture Notes in Mathematics. Springer, 2011.

[2] T. Athawale and A. Entezari. Uncertainty quantification in linear interpolation for isosurface extraction. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2723–2732, Dec 2013.

[3] Ken Brodlie, Rodolfo Allendes Osorio, and Adriano Lopes. A review of uncertainty in data visualization. In *Expanding the Frontiers of Visual Analytics and Visualization*, pages 81–109. Springer, 2012.

[4] Faruk H. Bursal. On interpolating between probability distributions. *Applied Mathematics and Computation*, 77(2-3):213 – 244, 1996.

[5] Kerry Emanuel. Increasing destructiveness of tropical cyclones over the past 30 years. *Nature*, 436(7051):686–688, 2005.

[6] P. Goovaerts. *Geostatics for Natural Resources Evaluation*. Applied geostatistics series. Oxford University Press, 1997.

[7] H.-D. Hollweg, U. Böhm, I. Fast, Hennemuth B., K. Keuler, E. Keup-Thiel, M. Lautenschlager, S. Legutke, K. Radtke, B. Rockel, M. Schubert, A Will, M. Woldt, and C. Wunram. Ensemble simulations over europe with the regional climate model clm forced with ipcc ar4 global scenarios. technical report, 2008.

[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, September 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, October 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] Richard J Larsen and Morris L Marx. *An introduction to mathematical statistics and its applications; 5th ed.* Prentice Hall, Boston, MA, 2012. The book can be consulted by contacting: IT-ESDNG: Abler, Daniel.

[12] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *COMPUTER GRAPHICS*, 21(4):163–169, 1987.

[13] Alex T. Pang, Craig M. Wittenbrink, and Suresh K. Lodh. Approaches to uncertainty visualization. *The Visual Computer*, 13:370–390, 1996.

[14] T. Pfaffelmoser, M. Mihai, and R. Westermann. Probability distributions for gradient orientations in uncertain 3d scalar fields. Technical report, Technische Universität München, 2012.

[15] 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.

[16] Kai Pöthkow and Hans-Christian Hege. Positional uncertainty of isocontours: Condition analysis and probabilistic measures. *IEEE Transactions on Visualization and Computer Graphics*, 17(10):1393 – 1406, 2011.

[17] Kai Pöthkow and Hans-Christian Hege. Nonparametric models for uncertainty visualization. *Computer Graphics Forum*, 32(3):131 – 140, 2013.

[18] Kai Pöthkow, Christoph Petz, and Hans-Christian Hege. Approximate level-crossing probabilities for interactive visualization of uncertain isocontours. *International Journal for Uncertainty Quantification*, 3(2):101 – 117, 2013.

[19] Kai Pöthkow, Britta Weber, and Hans-Christian Hege. Probabilistic marching cubes. *Computer Graphics Forum*, 30(3):931 – 940, 2011.

[20] Carl E. Rasmussen and Christopher Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[21] S. Schlegel, N. Korn, and G. Scheuermann. On the interpolation of data with normally distributed uncertainty for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2305–2314, 2012.

[22] Steven Schlegel, M. Böttinger, Mario Hlawitschka, and Gerik Scheuermann. Determining and visualizing potential sources of floods. In *EuroVis Workshop on Visualisation in Environmental Sciences*, Leipzig, 2013.

[23] Steven Schlegel, Mathias Goldau, and Gerik Scheuermann. Interactive gpu-based visualization of scalar data with gaussian distributed uncertainty. In David Bommes, Tobias Ritschel, and Thomas Schultz, editors, *VMV 2015 - Vision, Modeling and Visualization*. Eurographics Association, 2015.

[24] J. Sillmann and E. Roeckner. Indices for extreme events in projections of anthropogenic climate change. *Climatic Change*, 86(1-2):83–104, 2008.