Master Thesis

Development of anisotropy estimation methods using data from sensor networks

Spiliopoulos Ioannis

Department of Electronics & Computer Engineering

Technical University of Crete

Octomber 2010

Advisory Committee:
Prof. Zervakis Michalis[1]
Prof. Liavas Athanasios[1]
Prof. Hristopulos Dionisis[2]

[1] Department of Electronics and Computer Engineering of T.U.C.
[2] Department of Mineral Resources Engineering of T.U.C.

Abstract

This thesis addresses the estimation of geometric anisotropy parameters from scattered spatial data obtained from environmental surveillance networks. Environmental monitoring applications aim to provide real-time maps of air, ground and water pollution in an automatic way. Maps illustrate where pollution is coming form and where it is headed. Such information enables public authorities to decide more quickly on appropriate action. Estimates of geometric anisotropy improve the accuracy of spatial interpolation procedures that aim to generate those maps. The anisotropy parameters in two dimensions involve the orientation angle of the principal anisotropy axes and the anisotropy ratio (i.e., the ratio of the principal correlation lengths). The approach that we employ is based on the Covariance Hessian Identity (CHI) method [Hri02, CH08], which links the mean gradient tensor with the Hessian matrix of the covariance function [Swe62]. We extend CHI to clustered CHI (CCHI) for application in scattered data that include patches of extreme values and clusters of varying sampling density. We compare the performance of CHI and CCHI by means of synthetic datasets that involve different spatial distributions and demonstrate the importance of segregation for scattered data. We compare the performance of CCHI and various unsupervised clustering algorithms with respect to anisotropy estimation. The proposed clustering method CCHI marginally outperforms the competition. In addition, CCHI has minimum parameter requirements. We investigate the impact of CHI anisotropy estimation on the performance of spatial interpolation by means of ordinary kriging using a data set that involves both real background radioactivity measurements and a simulated release of a radioactive plume. Finally we discuss and briefly examine the application of CHI in combination with a moving window procedure to derive local estimates of anisotropy. The motivation for this extension of *CHI* is its application to Magnetic Resonance Imaging data.

Acknowledgements

I would like to express my sincere gratitude to my advisor, Professor Zerbakis Michalis for his inspiration and guidance. A very special thanks goes out to Professor Hristopulos Dionisis. It was under his tutelage that I developed an interest to anisotropy estimation methods and managed to complete my master thesis. I would also like to thank Professor Liavas Athanasios for being part of my advisory committee.I must also acknowledge Ersi Chorti whose work this thesis continues.

I would also like to thank my family for their support. This thesis is dedicated to them.

Outline

Introduction A general introduction about geostatistics and its possible applications

Classic Geostatistics In the present chapter we examine some of the fundamental properties of Random Fields (RF), present the main concepts of RF anisotropy, and discuss its classic estimation via directional variograms. We introduce conditions required for the existence of a random field. Properties such as stationarity, differentiability and ergodicity are briefly discussed. Moreover, the main types of anisotropy presented in the geostatistical literature are shortly described in terms of directional variograms.

Inference Methods In this chapter we present some of the inference methods for anisotropy parameter estimation and unsupervised clustering. Three methods presented in this section are used to investigate the data sets in chapters 5 and 6. First, the Covariance Hessian Identity (*CHI*) method is briefly described. The *CHI* method is at the core of the Clustered *CHI* algorithm which is presented in the next chapter. Second, three different clustering methods such as k-means, x-means and DBSCAN are described. These clustering algorithms are used and compared in chapters five and six [ch. 5,ch. 6]. Some of them could be used instead of the clustering algorithm presented in *CCHI* under in certain cases.

Clustered CHI Anisotropy Estimation method-*CCHI* This chapter presents in detail the methodologies proposed for anisotropy estimation. The procedures involved focus on the following tasks: (i) segmentation of the sensor network in domains of normal and extreme values (ii) subsequent partitioning of each domain into clusters of similar sampling density (iii) estimation of anisotropy parameters in each cluster and (iv) aggregation of the cluster parameters into coarse-grained anisotropy parameters valid in each domain. The procedures are described below using the GDR data set for illustration.

Anisotropy Analysis of Synthetic Data In this chapter we will present synthetic data-sets used for the performance analysis of *CHI* and *CCHI* algorithms. The synthetic scenarios are realizations of Gaussian Random Fields (GRF) generated via the Fast Fourier Transform methodology described later in subsection 1. In addition some of the on-grid GRFs are sub-sampled in order to generate scatter data-sets. Several realizations of these scattered

data-sets are combined in order to generate clusters with various anisotropy parameters as presented in subsection 2. Synthetic scattered sensor networks are used to validate *CCHI* algorithm performance.

Anisotropy Analysis of Real Data Two different real-case scenarios provided by the *European Radiological Exchange Platform* (EURDEP) Gamma Dose Rate (GDR) network are described later in this chapter. First, the *rain-event* scenario which monitors radioactivity over the European continent during episodes of heavy rainfalls. Second, *worst-case* scenario monitors the GDR over Europe several hours after a simulated nuclear accident at a nuclear facility in Belgium. The performance of *CCHI* is compared to other clustering algorithms in terms of anisotropy parameter estimation. In addition, a cross- validation analysis for the accuracy of the worst case scenario is presented here, along with prediction maps generated by means of kriging.

Application of CHI method on MRI data In this chapter we investigate the application of *CHI* on Magnetic Resonance Imaging (MRI) data. Nowadays medical imaging applications increasingly attract the interest of the scientific community. MRI data have proved useful for imaging brain diseases. In the following chapter, we present some of the basic data types used in the literature and their connection to anisotropy estimation. We discuss the application of *CHI* to various data types. Finally we present application of *CHI* on MRI data, by means of a moving window procedure, to provide local estimates of water concentration anisotropy . Water concentration in the brain structure is highly connected to the diffusion of cancer cell's, therefor local estimates of anisotropy may provide useful information on tumor modeling. Methods developed in Geostatistics aim to solve problems dealing with the characterization of spatial and spatio-temporal phenomena. Most of these phenomena emerge in scientific fields as mining, hydrology, meteorology, oceanography and environmental monitoring systems. On one hand, most observations are distributed over macro or earth-scale domains. On the other hand, the underlying scale is not a requirement for applying Geostatistics.

Conclusions Conclusions

Appendix A Some preliminary work about extension of *CHI* on three dimensions

Appendix B Tutorial on Anisotropy estimation using R source code that we developed for the Intamap R package.

Appendix C Help files for the R source code function that exist in Intamap R package

Contents

Table 1: **Notation**

Symbol	Explanation
$\otimes$	Kronecker product
$\|\| \cdot \|\|$	Norm of vector
$\| \cdot \|$	Absolute value of scalar
$E[\cdot]$	Expected value
∂	Partial derivative
∇	Natural gradient
$\tilde{x}$	Frequency space of x
$\hat{x}$	Sample average of x
$\overline{x}$	Weighted sample average of x
Ω	Sample space
ω	Sample point $\in \Omega$
S	Location space, usually $S = \mathbb{R}^2$
$\mathbb{R}^n$	N-dimensional euclidean Space
$Prob\{\cdot\}$	Probability of an event
$\mathbf{s}_i$	Position, vector at location i
$\mathbf{r}_{ij}$	Distance $\|\|\mathbf{s}_i - \mathbf{s}_j\|\|_2$ in original space
$\mathbf{h}$	Distance $\mathbf{s}_i - \mathbf{s}_j$ in transformed space
$\mathbf{x}$,$X(s_1, s_2)$	Vector $\mathbf{x}$,Value of X at location(s_1, s_2)
$\mathbf{X}$	Multi-dimensional matrix
$\mathbf{X}^T$	Transpose of $\mathbf{X}$
$X(\mathbf{s})$	Random field, process
$X(\mathbf{s}, \omega)$,X_s	Realization of X
$R_{(l)1}$	Ratio of correlation lengths l over 1
θ	Angle between major correlation length and x axes
C_{x}	Covariance matrix of process $X(\mathbf{s})$
H	Hessian matrix
$\tilde{X}(\mathbf{f}), \tilde{C}(\mathbf{k})$	Spectral Density and Covariance Function
$Q_{ij}, \hat{Q}_{ij}$	Average sample derivatives and its estimate over i and j direction
$\hat{Q}_{ij}^{g;c}$	Equal to $\hat{Q}_{ij}$ for cluster c which is part of domain g
$\overline{Q_{ij}^{g}}$	Weighted average over domain g of cluster slopes $\hat{Q}_{ij}$
$w_{g;c}$	Weight of cluster c for $\overline{Q_{ij}^{g}}$ estimation over domain g
L^2	# of nodes on rectangular lattice
N_c	# of sensor locations inside the cluster c
N_g	# of sensors inside domain g
N_t	# of sensors used as Training set
N_v	# of sensors used as Validation set

Table 2: **Abbreviations**

Abbreviation	**Explanation**
RF	Random Field
GRF	Gaussian Random Field
SRF	Stationary Random Field
BLUP	Best Linear Unbiased Predictor
CTI	Covariance Tensor Identity
CHI	Covariance Hessian Identity
CCTI	Clustered Covariance Tensor Identity
CCHI	Clustered Covariance Hessian Identity
GKP	Gradient Kronencker Tensor Product
SDG	Sampling Density Grid
SDM	Sampling Density Matrix
SSD	Similar Sampling Density (refer to clusters)
FFT	Fast Fourier Transform
IFFT	Inverse Fast Fourier Transform
et al	and others
wrt	with respect to
i.e	"id est", that is
AD	**Anisotropic** covariance function on multiple-cluster **Domain** analysis
ID	**Isotropic** covariance function on multiple-cluster **Domain** analysis
AS	**Anisotropic** covariance function on **Single** cluster analysis
IS	**Isotropic** covariance function with **Single** cluster analysis

Chapter 1

Introduction

Accuracy has always been the goal in all scientific fields that involve estimation and prediction problems. From the statistical point of view, accurate estimation is translated into finding the best predictor of an unknown random variable of scientific interest.

In spatial statistics, the optimum linear solution to this problem is provided by the Best Linear Unbiased Predictor (BLUP). BLUP has been developed independently in different scientific fields. Hence, BLUP is often given different names, such as Gaussian process regression or Kolmogorov-Wiener prediction. In geostatistics the spatial BLUP is known as Kriging. Various researchers in the early 60s developed methods similar to Kriging. According to Cressie [Cre90] only the independent studies of Matheron in France [Mat63] and Gandin in Russia [Gan63] incorporate all the components that define the method known today as Kriging.

The main difference between Kriging and the original BLUP, presented earlier by Yaglom in [Yag62], is the dependence of the covariance function on the spatial locations of the data. In addition, given a spatial process $X(\mathbf{s})$, this question arises: How does $X(\mathbf{s})$ change over space? Anisotropy partly answers this question. Anisotropy describes how a spatial process changes along different directions. Ecker in [EG99] states that, when modeling the correlation structure of a spatial process $X(\mathbf{s})$, the assumption of identical properties in all directions (isotropy) is not always valid. Thus, anisotropy has to be taken into account when modeling spatial processes.

There are various approaches in bibliography related to anisotropy parameters estimation. Some are based on Exploratory Data Analysis (EDA) techniques. Important examples of the EDA approaches are those presented by Kaluzny and Isaaks. Kaluzny in [KVCS96] proposed that directional semi-variograms can model departure from isotropy. In the same context, Isaaks [Hoh91] proposed that a rose diagram or a contour plot of empirical

variograms can give the same results. These EDA approaches do not directly yield parameter values. The parameters are chosen either based on visual inspection or by fitting the primary results (i.e. an empirical variogram) to a parametric model. An alternative is the Maximum Likelihood Estimation (MLE) approach [Kit83, Kit87, PI98] which seeks optimal parameters for a given covariance model conditioned by the data. However, multiple models have to be examined in order to achieve an optimum fit using MLE. Ecker in [EG99] approaches anisotropy from a Bayesian perspective. The posterior distribution of anisotropy parameters is estimated using a presumed prior distribution and non iterative Monte Carlo sampling from an importance sampling density distribution. The recently proposed Covariance Hessian Identity (*CHI*) method, which is a non-parametric and non-iterative method that applies to differentiable normal and lognormal random fields [Hri02, CH08]. In two dimensions *CHI* provides anisotropy estimates related to the spatial derivatives of the random field through closed forms. Serious advantages of *CHI* are that no prior information is needed and that it is computationally efficient. These attributes make *CHI* very attractive for automated procedures.

Anisotropy is a statistical property commonly used in geostatistics since the origins of the field. Examples of geostatistical techniques that are applied to other signal processing fields are rare in the literature. There are, though, applications such as image processing and medical imaging that can take advantage of techniques developed for spatial statistics. In this thesis we concentrate on geostatistics, but we also investigate briefly, whether the *CHI* anisotropy estimation method has potential benefits for medical imaging applications.

The core issue addressed by this work is the estimation of the anisotropy parameters $[\hat{R}, \hat{\theta}]$ of a random field (RF) $X(\mathbf{s})$ in two dimensions. Preliminary research on three dimensions is also presented. In our case $X(\mathbf{s})$ describes the Gamma Dose Rate distribution over Europe. In practice, there is only a finite set of monitoring locations. An additional difficulty is that the observations of a monitoring network are usually placed at the nodes of an irregular lattice. In this work we propose separation of the study area into smaller areas. Separation based on similar sampling density (SSD) of the sensor network increases the accuracy of *CHI* . Accurate θ and R estimates can then be incorporated in the covariance function used in kriging interpolation and hence improve the resulting prediction maps.

We also present a first attempt of anisotropy estimation in water concentration patterns, obtained using Magnetic Resonance Imaging (MRI). Even though the MRI data are sampled on micro scale, they exhibit spatial dependence similar to problems that occur in Geostatistics. Anisotropy esti-

mation on MRI data is important for the modeling of disease processes such as brain tumors. CHI application may lead to important advances in this scientific field. Existing methods, such as fractional anisotropy (FA) and relative anisotropy (RA), can only point out departure from isotropy. On the other hand, the existing implementation of CHI provides estimates that fully model geometric anisotropy in twodimensions (i.e. it provides the ratio of correlation semi-axes and the direction of the major correlation axis).

Preliminary research presented in this thesis aims to solve the equations of *CHI* in three dimensions. The solution of *CHI* in three dimension can provide important benefits for medical imaging. To date, the numerical minimization approach that is used is trapped in local minimums. Closed form solution of the thee dimensional equations is provided in the special case where one of the anisotropy principal axes is aligned with the vertical axis of the coordinate system.

Chapter 2

Classic Geostatistics

In the present chapter we examine some of the fundamental properties of Random Fields (RF), present the main concepts of RF anisotropy, and discuss its classic estimation via directional variograms. We introduce conditions required for the existence of a random field. Properties such as stationarity, differentiability and ergodicity are briefly discussed. Moreover, the main types of anisotropy presented in the geostatistical literature are shortly described in terms of directional variograms.

1 Random Fields

In this section we present the basic principles related to the theory of random fields. A major influence is the technical report [Abr97] that we follow through this section. We examine the basic properties that a set of finite-dimensional functions must admit in order to be valid covariance models. We discuss properties such as stationarity, continuity, separability, differentiability and positive-definiteness.

One may find several descriptions or informal definitions of Random Fields (RF). In most cases, these descriptions pass step by step through *random variable* to *random process* and then generalize to multi - space random processes called *Random Fields*. The definition of random fields often comes with its analogous definition of random variables, with respect to an arbitrary scientific experiment. Such a definition is found in [Van88]:
" *Random Variable is the outcome, that incorporates some uncertainty, of a procedure or experiment designed to discover an unknown truth or effect. In this sense Random Fields may be described as the total outcome of a very large number of such experiments.*"

Random Fields differ from *Random Processes* only in the fact that their parameters are defined over a spatially correlated multi - dimensional space. So a *Random Field* is a stochastic model that is used to describe certain types of spatial processes, such as those that often occur in geostatistics [Bes74].

1.0.1 Definition of Random Fields

As stated in [Abr97] the study of a random field usually implies the study of its covariance function. Most of the properties of a *Random Field* are analogous to those of *random processes*. A function is a valid covariance function only if it is positive-definite and continuous. However, in *Random Fields* the covariance and correlation function are modified to incorporate some geometric or spatially dependent properties. Such a property is isotropy. The covariance functions of *Random Fields* take into account the spatial dependence in the set of sampling locations.
The formal definition of *Random Fields* is given in [Abr97] as

Definition 1. ***Random Field*** *Let a probability space, (Ω, F, P), and a parameter set, S, be given. A random field is then a finite or real valued function $X(\mathbf{s}, \omega)$ which, for every fixed $s \in S$ is a measurable function of $\omega \in \Omega$*

We note here that for fixed $\omega \in \Omega$, $X(\mathbf{s}, \omega)$ is a non-random function of s (*called position or coordinate*) that is denoted x_s (*called realization or sample function*). In case of an n-dimensional Euclidean space $S = \mathbb{R}^n$ with $n \geq 2$[1] we will denote the random field as

$$X_{\mathbf{s}} := X(\mathbf{s}, \omega), \quad s \in \mathbb{R}^n.$$

The *expectation* of a random field is by definition:

$$m(\mathbf{s}) = E[X_{\mathbf{s}}] = \int_{\Omega} X(\mathbf{s}, \omega) dP(\omega),$$

which may be expressed by using the finite dimensional distribution $F_{\mathbf{s}}(x)$ in $\mathbb{R}^1$ as follows :

$$m(\mathbf{s}) = \int_{\mathbb{R}^1} x \; dF_{\mathbf{s}}(x).$$

[1] The dimension of the coordinate system is valid for $n \geq 0$, but it is usually in the range of one to four. Also for $n = 1$ it is usually called a *stochastic or random process*. The term *Random Fields* implies that dimension is higher than one.

The covariance function is correspondingly expressed as

$$C(\mathbf{s}_i, \mathbf{s}_j) = Cov\{X_{\mathbf{s}_i}, X_{\mathbf{s}_j}\} = E[X_{\mathbf{s}_i} X_{\mathbf{s}_j}] - m(\mathbf{s}_i)m(\mathbf{s}_j)$$
$$= \iint_{\mathbb{R}^2} xy \; d^2 F_{\mathbf{s}_i,\mathbf{s}_j}(x, y) - m(\mathbf{s}_i)m(\mathbf{s}_j)$$

If the covariance function exists, then the probability density function (pdf) is obtained from the partial derivatives as

$$f_{s_1,\ldots\mathbf{s}_k}(x1, \ldots, x_n) = \frac{F_{s_1,\ldots,s_k}(x_1, \ldots, x_k)}{\partial x_1, \ldots, \partial x_k}.$$

In this case, both expectation and covariance function can be expressed in terms of probability density function as

$$m(\mathbf{s}) = \int_{\mathbb{R}^1} x f_{\mathbf{s}}(x) dx,$$
$$C(\mathbf{s}_i, \mathbf{s}_j) = \iint_{\mathbb{R}^2} xy \; f_{\mathbf{s}_i,\mathbf{s}_j}(x, y) dx dy - m(\mathbf{s}_i)m(\mathbf{s}_j).$$

1.0.2 Gaussian random fields

For the special category of Gaussian Random Fields (GRF), the joint probability density function can be fully described using the mean value m and covariance function C.

$$f_x(\mathbf{x}) = \frac{1}{2\pi^{(N/2)}|C_{\mathbf{x}}|^{1/2}} \exp\{-\frac{1}{2}(\mathbf{x} - \mathbf{m}_{\mathbf{x}})^T C_{\mathbf{x}}^{-1}(\mathbf{x} - \mathbf{m}_{\mathbf{x}})\}$$

The formal definition of Gaussian random fields is

Definition 2. *A Gaussian random field is a random field where all the finite-dimensional distributions, $F_{s_1,\ldots,s_k}$, are multivariate normal distributions for any choice of k and $s_1, \ldots, s_k$.*

1.0.3 Existence of Random Fields

A random field is usually described by its *finite-dimensional (cumulative) distributions* :

$$F_{s_1,\ldots,s_k}(x_1, \ldots, x_k) = \text{Prob}\{X_{s_1} \leq x_1, \ldots, X_{s_k} \leq x_k\}$$

The study of these distributions is related to the existence of a random field. The tool that someone must use for this purpose is the Kolmogorov's Theorem. If its conditions are met, then it guarantees the existence of such a field.

Theorem 1. ***Kolomogorov's Existence Theorem****: If a system of finite-dimensional distributions, $F_1 \dots F_k$, satisfies both symmetry and compatibility conditions, then there exists on some probability space (Ω, F, P) a random field $X_s : s \in S$ having $F_1 \dots F_k$ as its finite-dimensional distributions.*

Symmetry Condition: Given a set of finite-dimensional distribution functions $F_{s_1,\dots,s_k}(x_1, \dots, x_k)$ and permutation π of the index set $\{1, \dots, k\}$ the following equation must hold:

$$F_{s_1,\dots,s_k}(x_1, \dots, x_k) = F_{s_{\pi 1},\dots,s_{\pi k}}(x_{\pi 1}, \dots, x_{\pi k}).$$

Compatibility Condition: Given a set of finite-dimensional distribution functions $F_{s_1,\dots,s_k}(x_1, \dots, x_k)$, the following equation must hold:

$$F_{s_1,\dots,s_{k-1}}(x_1, \dots, x_{k-1}) = F_{s_1,\dots,s_{k-1},s_k}(x_1, \dots, x_{k-1}, \infty).$$

This important theorem verifies the existence of a random field having this arbitrary set of dimensional distributions. However this does not mean that the existing field is unique. Kolomogorov provides a powerful theorem for inspection of random fields in theory. However, in practice, the interest for random fields usually comes with the inspection of continuity and differentiability of the realizations.

Positive definiteness is fundamental for all covariance functions. Its definition is given below:

Definition 3. *Let k be a positive integer, and let $s_i \in S$ and $c_i \in \mathbb{R}^1$ for $i = 1, \dots, k$. Then the function C on $S \times S$ is said to be positive (semi) definite on S if*

$$\sum_{i=1}^{k} \sum_{j=1}^{k} c_i c_j C(s_i, s_j) \geq 0$$

for any choice of $\{s_1, \dots, s_k\}$ and $\{c_1, \dots c_k\}$ for any positive integer k.

This property is fundamental. Its importance comes from the fact that a certain *positive definite function* is equivalent to the covariance function of the respected *random field*. This comes from the proof of the following theorem as presented in [Abr97].

Theorem 2. *The class of covariance functions coincides with the class of positive definite functions.*

1.1 Mathematical properties of random fields

1.1.1 Stationarity of random fields

Stationarity for any random function is a property very similar to *transition invariance*. This property ensures that the characteristics of a given function stay the same over any shifting transformation, of an arbitrary set of n points.

Definition 4. ***Stationarity in the strict sense:*** *A random field, is said to be* stationary in the strict sense *if for any arbitrary set of k points $\{s_1, \dots s_n\} \in S$ and for every random vector $s \in S$*

$$F_{s_1,\dots,s_k}(x_1, \dots, x_k) = F_{s_{1+s},\dots,s_{ks}}(x_1, \dots, x_k),$$

i.e a translation of a point configuration in a given direction does not change the multiple-joint distribution.

In practice this strict sense of stationarity is difficult to ascertain for every arbitrary set of points. Instead, it is usually enough to admit *wide sense stationarity* of a random field. *Wide sense stationarity* means that the first and second moments of the field do not change over any arbitrary set of points $\mathbf{s} \in S$.

Definition 5. ***Stationarity in the wide sense :*** *A random field is stationary in the wide sense if*

$$m(\mathbf{s}) = m \quad \textit{and} \quad C(\mathbf{s_i}, \mathbf{s_j}) = C(\mathbf{r_{ij}}), \quad \textit{where } \mathbf{r_{ij}} = \mathbf{s_i} - \mathbf{s_j}$$

Often in place of *wide sense stationarity* one may find the notion of *intrinsic stationarity*. The only difference is that *intrinsic stationarity* demands stationarity of the first two moments for the increment of an arbitrary pair of values.

In any case stationarity in the strict sense implies stationarity in the wide sense, whereas the opposite is not always true. On the other hand, for Gaussian Random Fields these conditions are equivalent [Abr97, Wac03].

1.1.2 Continuity, Differentiability, Separability and Ergodicity

Continuity As stated earlier, an important aspect of Random Fields is the continuity and differentiability of the *realizations or sample functions*. When we refer to the continuity of a random field, we indirectly refer to the convergence of the sequences $X(\mathbf{s}_n)$ of random variables at each location to a certain value. Given that different types of convergence exist for random variables, one may suspect more than one type of continuity for Random Fields. There are three types of continuity whose definitions follow

Definition 6. ***Continuity of random fields***
Consider $S \subset \mathbb{R}^n$

(i) A random field X *has continuous sample functions with probability one in* S *if for every sequence* $\{\mathbf{s}_n\}$ *for which* $||\mathbf{s}_n - \mathbf{s}|| \to 0$ *as* $n \to \infty$,

$$Prob\{\omega : |X(\mathbf{s}_n, \omega) - X(\mathbf{s}, \omega)| \to 0 \quad as\ n \to \infty \quad for\ all\ \mathbf{s} \in S\} = 1.$$

(ii) A random field X *is almost surely continuous in* S*, if for every sequence* $\{\mathbf{s}_n\}$ *for which* $||\mathbf{s}_n - \mathbf{s}|| \to 0$ *as* $n \to \infty$,

$$Prob\{\omega : |X(\mathbf{s}_n, \omega) - X(\mathbf{s}, \omega)| \to 0 \quad as\ n \to \infty\} = 1 \quad for\ all\ \mathbf{s} \in S$$

(iii) A random field X *is mean square continuous in* S*, if for every sequence* $\{\mathbf{s}_n\}$ *for which* $||\mathbf{s}_n - \mathbf{s}|| \to 0$ *as* $n \to \infty$,

$$E\{|X(\mathbf{s}_n) - X(\mathbf{s})|^2\} \to 0 \quad as\ n \to \infty \quad for\ all\ \mathbf{s} \in S$$

As one may notice, these different types of continuity do not all guarantee *continuous sample functions*. Only the first one admits, with "probability one," continuous sample functions. As the continuity of sample paths usually is more important or interesting in practice, it makes sense to include additional conditions that guarantee this. In [Doo53, Adl81] is stated that *separability* of the random fields implies continuity of the sample functions.

Separability For any random process, separability ensures that finite dimensional distributions determine sample function properties only by requiring that the sample functions' values define a countable subset of positions in $\mathbb{R}^n$. This does not apply to random fields [Abr97]. However in [Doo53] it is stated that it is always possible to find an equivalent separable random field X for any given random field Y. Given this statement one may assume that an equivalent random field with continuous sample functions always exists.

Ergodicity According to [Van88] a random field is *ergodic* if all the information about its joint probability distributions and the statistics can be obtained from a single realization of the field. In practice, ergodicity is important because only a single realization of the field is usually available.

1.1.3 Simulation of Random Fields

Several methods exist for random fields simulation.In this section we will discuss briefly some of the most commonly used approaches. Simulation of a random field aims to generate valid realizations of a specific random field. One way to achieve this is to generate all possible realizations, which is not always possible. A more efficient way is the Metropolis-Hastings algorithm that samples realizations according to their probability. There are two main types of simulations that generate random fields realizations: the **unconditioned** and the **conditional**. Unconditioned simulation takes into account only the statistics (such as mean and covariance) of the simulated field, while conditional simulation also considers the measurements at the sampling locations.

The main methods for unconditional simulation include LU decomposition, turning bands, Harmonics Superposition and Fast Fourier Transform. We will use the latter to generate synthetic data. On the other hand, for conditional simulation we use the LU decomposition along with a measurements vector. A combination of unconditioned simulation with Kriging interpolation is also used. Finally we use the Metropolis algorithm combined with simulation annealing.

2 Variogram

Like the covariance, the variogram function is a useful tool for describing the variation in space of a random function and in respect of a random field [Wac03]. Given a random function $X(\mathbf{s})$, the variation of the field can be described in terms of pairwise differences between points such as $x(\mathbf{s}_i)$ and $x(\mathbf{s}_j)$. These differences

$$X(\mathbf{s}_i) - X(\mathbf{s}_j)$$

are called *increments*. Looking at the definition of the theoretical variogram below one may notice some relation to the variance of the increments. Ideally, the expected value of increments is zero. These two relations lead to the definition of second order stationarity for the *increments*, and in respect to that, *intrinsic stationarity* for the random field $X(s)$:

$$\gamma(r_{ij}) = \frac{1}{2}E[(X(\mathbf{s}_i) - X(\mathbf{s}_j))^2].$$

In any case, a constant mean or variance is not necessary in intrinsic stationarity. The following parameters are often used to describe variograms:

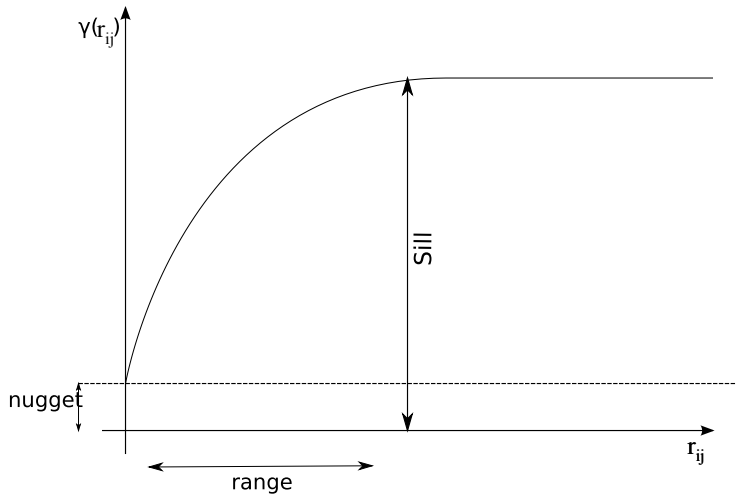


Figure 2.1: Theoretical Variogram

- nugget c_0: The height of the jump discontinuity of the semi-variogram at the origin.

- sill c_1: Limit of the variogram value as lag distances approach infinity.

- range r_c: The distance in which the difference of the variogram from the sill becomes negligible.

In variogram models with a fixed nugget c_0, it is the distance at which the sill is reached; for models with an asymptotic sill, it is conventionally taken to be the distance when the semi-variance first reaches 95% of the sill.

Experimental Variogram In practice, we use the experimental variogram to estimate the variogram. Given a data set S, we calculate the pairwise distance vector $\mathbf{r_i}$:

$$\mathbf{r}_{ij} = ||\mathbf{s}_i - \mathbf{s}_j||_2 \quad \forall \quad \mathbf{s}_i, \mathbf{s}_j \in S$$

From the set of vector $\mathbf{r_{ij}}$, only the distances that are are approximately equal an element of vector $\mathbf{h}$ will be used. The vector $\mathbf{h}$ is the separation set. The separation set includes the set of distances that is chosen to calculate the experimental variogram. The distances that belong to $\mathbf{h}$ are called *lags*.

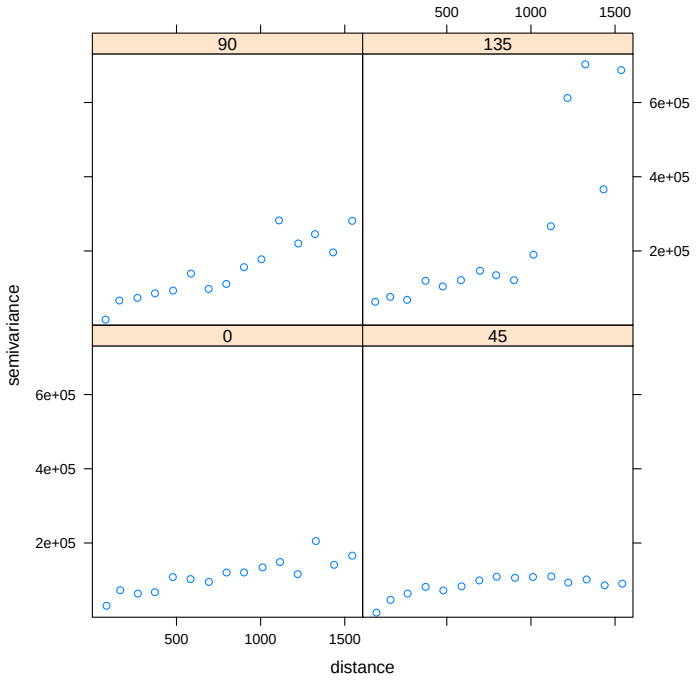


Figure 2.2: Directional Variogram in directions 0°,45°,90° and 135°from horizontal axis of Zn observations measured in the top soil in a ood plain along the river Meuse. The different sills along different directions indicate zonal anisotropy.

Hence for the k_{th} *lag* we use only those pairs whose distance is approximately equal to the h_k:

$$S_k = \{\mathbf{s}_i, \mathbf{s}_j \quad | \quad r_{ij} \approxeq h_k\}.$$

Denoting the number of those pairs by N_k, we can write the variogram method of moments estimator as follows [Mat63]:

$$\hat{\gamma}(h_k) = \frac{1}{2N_k} \sum_{i,j \in S_k} (X(\mathbf{s}_i) - X(\mathbf{s}_j))^2.$$

In respect to an experimental variogram definition, the **directional variogram** uses only those pairwise distances that are aligned with a specified direction and tolerance.

3 Anisotropy

Anisotropy is the opposite of isotropy, which implies that a certain property has the same behavior across all directions. The etymology of the word *"anisotropic"* comes from the combination of the ancient Greek words *"anisos" and "tropos."* The first word means "unequal" and the second one means "in a way."

As Clark stated in [Cla79], the form of anisotropy seen most often in practice is that of direction-dependent range. When speaking about range, one should have in mind the classic approach of anisotropy detection, the directional semi-variograms. When the combined plot of directional experimental semi-variograms, or the iso-variogram lines, indicates that the range varies with direction, without having similar behavior for the sill and the nugget, then we may say that this behavior yields geometric or range anisotropy as defined in [Zim93].

Geometric Anisotropy The current practice in modeling geometric anisotropy with semi-variograms [Hoh91] is to estimate subjectively the axes of anisotropy and the degree of anisotropy (i.e. the ratio of the maximum range to the minimum range) by visual inspection of the directional experimental variograms or rose diagram; then based on these estimates, to transform the coordinate system to achieve isotropy; and finally to fit an anisotropic model to the omni-directional experimental semi-variogram that is recalculated in the new coordinate system.

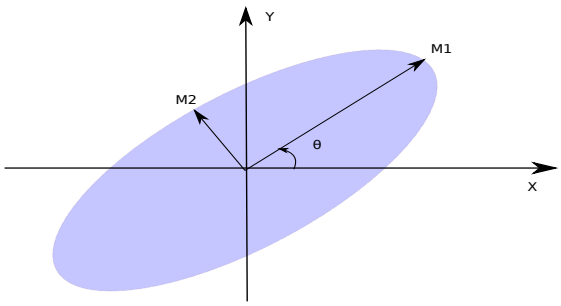


Figure 2.3: The coordinate system $\vec{h} = (x, y)$ and the major axes of anisotropy $\vec{h'} = (M1, M2)$

In a 2-D space, a visual representation of geometrical anisotropy of random fields may be approximated by concentric ellipses that approach the iso-variogram lines of the examined random field. In the special case where the iso-variogram lines are circular, the random field is isotropic. The geometric anisotropy can be obtained by a linear transformation of spatial coordinates or locations of the corresponding isotropic model. In 2D, given the vector of coordinates $\mathbf{s} = (\mathbf{x}, \mathbf{y})$ and a Rotation matrix U, we may transform our data based on

$$\mathbf{s}' = \mathbf{U}\mathbf{s}$$

where U is

$$U = SR = \begin{bmatrix} 1 & 0 \\ 0 & R^2_{1(2)} \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}.$$

Finally the transformed coordinates are linked to the original ones by means of:

$$\begin{bmatrix} \vec{x}' & \vec{y}' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -R^2_{1(2)}\sin\theta & R^2_{1(2)}\cos\theta \end{bmatrix} \begin{bmatrix} \vec{x} \\ \vec{y} \end{bmatrix}.$$

Zonal and other types of anisotropy Zonal anisotropy can be viewed as the representation of two processes, one of which has a lower-dimensional support. Besides geometric anisotropy, several other types of anisotropy have been proposed; with zonal anisotropy to be the one that is most frequently used. Other types of anisotropy are often derived in a way that explains a certain behavior of variograms (*sill* and *range* anisotropy as defined from Zimmerman in [Zim93]); however most of these types can be incorporated in the spatial fields model as a trend, leaving only geometric anisotropy to be modeled separately.

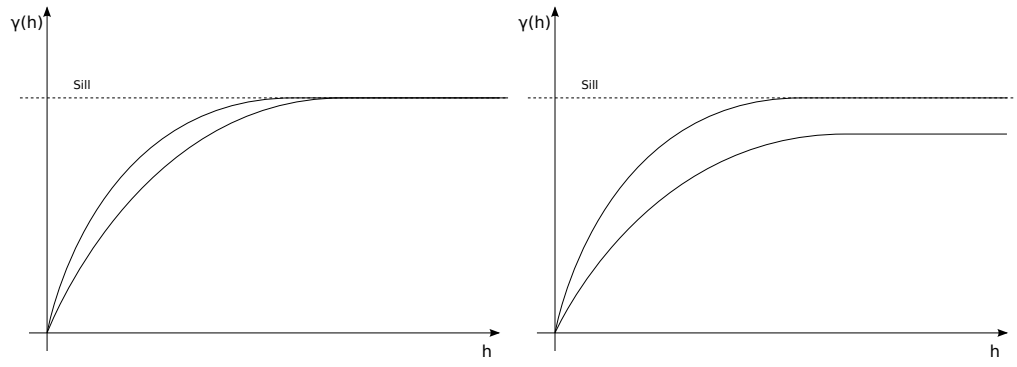


Figure 2.4: Left variogram showing geometric anisotropy. Right variogram showing zonal anisotropy.

Chapter 3

Inference Methods

In this chapter we present some of the inference methods for anisotropy parameter estimation and unsupervised clustering. Three methods presented in this section are used to investigate the data sets in chapters 5 and 6. First, the Covariance Hessian Identity (*CHI*) method is briefly described. The *CHI* method is at the core of the Clustered *CHI* algorithm which is presented in the next chapter. Second, three different clustering methods such as k-means, x-means and DBSCAN are described. These clustering algorithms are used and compared in chapters five and six [ch. 5,ch. 6]. Some of them could be used instead of the clustering algorithm presented in *CCHI* under in certain cases.

1 The Covariance Hessian Identity (*CHI*) method

In this section we briefly describe the Covariance Tensor Identity (*CHI*) method for anisotropy detection; presented in more depth in [Hri02, CH08]. This method differs from classical approaches ,e.g. directional variograms [Wac03, Zim93] and maximum likelihood, for identification of anisotropic correlations in Spatial Random Fields, by means of a non-parametric and non-iterative procedure.

The *CHI* method is a non parametric and non-iterative method that applies to differentiable random fields with normal or log-normal probability density functions. This relatively new approach is based on sample based estimates of the random field spatial derivatives that are related to the anisotropy parameters through closed form mathematical expressions. A fundamental argument for the validity of this approach is the *Covariance Hessian Identity*

introduced by [Swe62] that links the second order derivatives of the covariance function to the expected value of the first order derivatives of the field. In [CH08], it is stated that assuming ergodic conditions, the covariance function can be estimated by means of suitable sample averages. As implied here, *CHI* assumes stationarity and existence of the second order derivatives for the covariance function.

Given that the SRF $X(s)$ is differentiable, the Covariance Hessian Matrix is

$$H_{ij} = -\frac{\partial^2 C_x(r)}{\partial r_i \partial r_j} \tag{3.1}$$

For normal and log normal random fields the existence of the first order field derivative is ensured by the existence of the second order derivative of the covariance function at zero lag. As defined in [CH08], the Gradient Tensor Product (GKP) is:

$$X_{ij} = \nabla X(s) \otimes \nabla X(s)^T = \partial_i X(s) \partial_j X(s), \tag{3.2}$$

and its expected value

$$Q_{ij} = E[X_{ij}] = E[\partial_i X(s) \partial_j X(s)]. \tag{3.3}$$

Based on [Swe62], one may write the *Covariance Hessian Identity* as

$$Q = H(r)|_{r=0} \tag{3.4}$$

In the same article it is shown that, after differentiation of the Hessian matrix at zero lag, equation (3.3) can be re-written using *Einstein notation* [1] in following form:

$$Q_{ij} = -\frac{R_{l(1)}^2}{\xi_1^2 d} \triangle c_x(0) U_{li}(\boldsymbol{\theta}) U_{lj}(\boldsymbol{\theta}) \quad \texttt{i,j=1...d} \tag{3.5}$$

where $R_{l(1)}$ expresses the ratio between correlation length ξ_l and correlation length ξ_1 and is equal to $R_{l(1)} = \frac{\xi_l}{\xi_1}$. In addition, $U_{ij}(\boldsymbol{\theta})$ are the elements of the rotation matrix in d-dimensions. The parameter vector of angle $\boldsymbol{\theta}$ defines the rotation angles for the rotation matrix U and dimension d of the random field. The remaining term

$$\triangle c_x(0) = \sum_{i=1}^{d} \frac{\partial^2 \tilde{c_x}(0)}{\partial r i^2}$$

[1] According to *Einstein notation*, when an index appears twice in a single term, it implies that we are summing over all its possible values.

is the Laplacian of the reduced isotropic covariance function. This term is eliminated during the estimation procedure. An explicit description and the proof of the previous functions can be found in [CH08].

As we stated earlier in this section, *CHI* is valid in theory for any finite number of dimensions $d \geq 2$. Assume that the slope tensor $\hat{Q_{11}}$ is the GKP element of highest value. Then it is proposed to cast the existing system of equations in terms of ratios of the CHM elements $H_{ij}(0)/H_{11}(0)$ and the respective sample slope tensors $\hat{Q}_{ij}(0)/\hat{Q}_{11}(0)$. In practice, on regular grids the sample slope tensors are estimated using discrete approximations of the first-order partial derivatives, e.g. on square grids of step a:

$$\hat{Q}_{ij} = \frac{1}{N}\sum_{k=1}^{N} \frac{X(\mathbf{s}_k + a\mathbf{e_i}) - X(\mathbf{s}_k)}{a} \frac{X(\mathbf{s}_k + a\mathbf{e_j}) - X(\mathbf{s}_k)}{a} \tag{3.6}$$

where $\mathbf{e_i}, \mathbf{e_j}$ are unit vectors in the respected directions.

In theory, these slope tensor ratios are equal; however, in practice, it is necessary to take into account both modeling and sampling errors. For this reason the residual of these terms is introduced as a function of the parameter vector

$$H_{ij}(0)/H_{11}(0) - \hat{Q}_{ij}(0)/\hat{Q}_{11}(0) = \epsilon_{ij}(\boldsymbol{\theta}, \mathbf{R_1}). \tag{3.7}$$

The anisotropy parameters may be determined by minimization of the following cost function $\Lambda(\{\hat{\boldsymbol{\theta}}; \hat{\mathbf{R}}_\mathbf{1}) = \sum_{i=2}^{d}\sum_{j\leq i}^{d} \epsilon_{ij}^2(\boldsymbol{\theta}, \mathbf{R_1})$ i.e.,

$$\{\hat{\boldsymbol{\theta}}; \hat{\mathbf{R}}_\mathbf{1}\} = \underset{\hat{\boldsymbol{\theta}}; \mathbf{R_1}}{\arg\min}\, \Lambda(\hat{\boldsymbol{\theta}}; \hat{\mathbf{R}}_\mathbf{1}) \tag{3.8}$$

In Two Dimensions the authors work directly with equation (3.5) for $d = 2$. In [CH08] a closed form solution for this problem is provided. In two dimensions, the slope tensors are $Q_{ij} \in \{Q_{11}, Q_{12}, Q_{22}\}$ and the corresponding ratios $(Q_{22}/Q_{11}, Q_{12}/Q_{11})$ are named $(q_{\texttt{diag}}, q_{\texttt{off}})$ respectively. The closed form solution is

$$\theta = \frac{1}{2}\tan^{-1}\left(\frac{2q_{\texttt{off}}}{1 - q_{\texttt{diag}}}\right) \tag{3.9}$$

$$R_{2(1)} = \sqrt{1 + \frac{1 - q_{\texttt{diag}}}{q_{\texttt{diag}} + (1 + q_{\texttt{diag}})\cos^2\theta}} \tag{3.10}$$

where $\theta \in [-\pi/2, \pi/2]$ and $R_2(1) \in [0, \infty)$.

In three Dimensions slope tensors may be written explicitly; however this problem is hard to solve in closed form, and the estimation is performed through minimization.

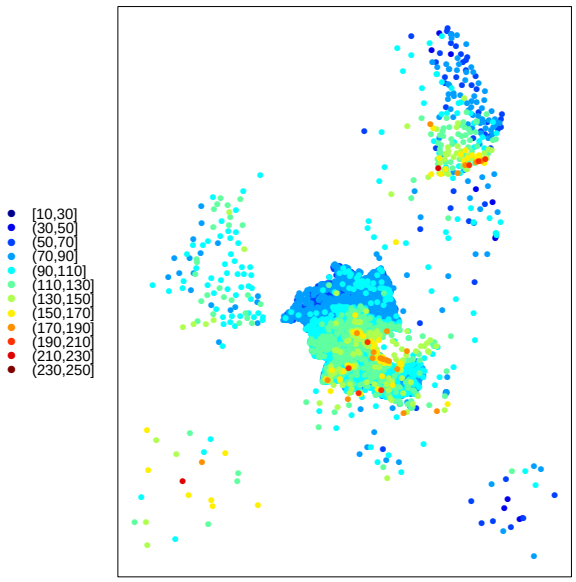


Figure 3.1: Instance of a working Gamma Dose Rate monitoring network over Europe.

2 Clustering-Segmentation Algorithms

In this section we give a brief description of some related work on clustering algorithms. The goal of this work has been to provide a methodology able to segment a monitoring sensor network in an almost-real time application. The goal is to increase *CHI* anisotropy estimation performance, in terms of accuracy, when dealing with large scatter datasets. Limitations to the specified methodology arise both from *CHI* method assumptions and the network structure itself.

A working sensor network has a transient geometry, as there are often missing or malfunctioning sensors (see fig.3.1). In addition, sometimes smaller networks are combined with nation-wide or even continent-wide networks. It is possible then to have missing regions that cover a significant percentage of the entire network area, due to synchronization errors or different reporting policies. Background *CHI* is designed to work on rectangular grids; for scattered data it takes advantage of interpolation to project the information onto the grid. Due to the statistical nature of the *CHI* method, it is preferable to not have very small clusters. This yields a limitation on the minimum number of sensors that comprise a sensor cluster.

2.1 K-means

There are various distance-based techniques;for unsupervised clustering of data. However a standard approach is *K-means* [RDD73]. K-means requires the user to specify in advance the parameter k that represents the number of clusters. Then k randomly chosen locations are initially selected as cluster centers. All samples are assigned according to Euclidean distance to the closest cluster centroid. These centroids are updated after each step. Each updated centroid is calculated as the mean of the samples assigned to the respective cluster. Then all distances from each centroid are re-calculated. This procedure is repeated until the centroids remain fixed.

2.2 X - means

X-means is an improved version of the widely used *k-means* algorithm. As described in [DP00], this new implementation solves to two of the three major problems with *k-means*, the computational scaling of the original algorithm and the need for prior knowledge of the number of clusters k.

The first problem is approached through a kd-tree implementation that stores sufficient statistics at its nodes; this approach speeds up the algorithm without approximations. The new improved version of *k-means* is a step in the procedure of *X-means*. Another step, the one that addresses the second problem, is the algorithm that estimates the number of clusters k. The estimation of k is performed after each iteration of the *k-means*. Then a subset of the current centroids is chosen to be split to improve the data fitting. The selection of the subset is based on the Bayesian Information Criterion (BIC) as presented in [KW95].

2.3 Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

The Density Based Spatial Clustering of Applications with noise (*DBSCAN*) algorithm was first presented by Ester et al in [EKSX96]. According to the authors, clustering algorithms of large spatial datasets often face three basic requirements: 1) Minimal requirements of domain knowledge to determine parameters. 2) Ability to discover clusters of arbitrary shapes. 3) Good efficiency on large databases.

The *DBSCAN* algorithm tries to meet these requirements. The chosen area of interest in each sample's vicinity is named here eps-neighborhood. The key idea of this algorithm is that the "density in the eps-neighborhood has to exceed a certain threshold." Two parameters are needed in advance:

the neighborhood range (*eps*) and the minimum number of points (m) inside this range. Any points that do not meet these criteria are treated as noise and remain unassigned.

As explained in [EKSX96] a naive approach to the algorithm would be to demand that each point of a cluster has inside its neighborhood a minimum number of points. However, in practice the algorithm demands that each point p of a cluster, has a neighboring q, that meets this limitation; this way it ensures that *border points* are not treated as noise. The point q is called *core-point*. Points (p, q) that meet this limitation (*eps* & m) are called *directly density-reachable*. In addition, if a chain of directly density-reachable points $p_1, \cdots, p_n$ from p to q exists, this pair is called *density-reachable*. Moreover if a point o exists that is *density-reachable* to both p and q, then the (p, q) pair is called *density-connected*. These definitions support the density-based notion of a cluster. Intuitively, a DBSCAN cluster is defined to be a set of density-connected points which is maximal with respect to density-reachability. The formal definition of a cluster as written in [EKSX96] follows:

Definition 7. *Let D be a database of points. A cluster C with respect to eps and m is a non-empty subset of D satisfying the following conditions:*

1. *$\forall p, q$: If $p \in C$ and q is density-reachable from p with respect to. eps and m, then $q \in C$. (Maximality)*

2. *$\forall p, q \in C$: p is density-connected to q with respect to eps and m. (Connectivity)*

The DBSCAN procedure can be summarized as follows: DBSCAN algorithm starts from an arbitrary point p and retrieves all *density-reachable* points from p. If p contains in its eps-neighborhood more than m points then a new cluster is formed. This cluster contains p and its density-reachable points. If p does not have enough neighbors it is considered temporarily as noise, and then the algorithm moves to a new point. In last case p might be assigned to a different cluster if it is density reachable from an other point. On one hand, DBSCAN tends to merge clusters of different density if those are close to each other by means of eps-neighborhood. On the other hand clusters of similar density may be separated from each other if the distances between clusters is larger than eps.

Chapter 4

Clustered CHI Anisotropy Estimation method

This chapter presents in detail the methodologies proposed in this thesis for anisotropy estimation. The procedures described below focus on the following tasks: (i) segmentation of the sensor network in domains of normal and extreme values; (ii) the subsequent partitioning of each domain into clusters of similar sampling density; (iii) the estimation of anisotropy parameters in each cluster; and (iv) the aggregation of the cluster parameters into coarse-grained anisotropy parameters valid in each domain. The procedures are described below using the GDR data set for illustration.

1 Clustered *CHI*

Consider an environmental sensor network (e.g., radioactivity probes) containing N sampling points $\mathbf{s}_i = (x_i, y_i)$, $i = 1, \ldots, N$, where (x_i, y_i) are the spatial coordinates on the globe expressed in an equidistant projection system. The sampled process is denoted by $X(\mathbf{s})$. We will assume that $X(\mathbf{s})$ is modeled by a *spatial random field* the realizations of which admit at least first-order partial derivatives. This class includes fields with Gaussian covariance, or Matérn covariance with smoothness index $\nu > 1$, or Spartan random fields with finite spectral cutoff $k_c < \infty$. The *CHI* method assumes that the data are generated from a second-order stationary (i.e., constant-mean and translation-invariant covariance function) random field with differentiable realizations.

Often, the stationarity assumption is not supported by the data. For example, in the GDR data described in Section 2, the radioactive release generates a plume whose statistical properties differ markedly from the background

radioactivity. To justify the stationarity assumption, we consider separately subsets of the sampling network that contain a large number (e.g., $N_g > 25$) of extreme values. Thus, separate *domains* are defined that contain the "normal" and "extreme" values respectively. Further, if the sampling density varies significantly over a domain, the latter is partitioned into clusters of *similar sampling density (SSD)*, using image segmentation methods based on edge detection [GW06]. In each cluster, we define a different anisotropy estimation grid by tuning the grid step to the cluster sampling density, as discussed in Section 1.1 below.

1.1 Segmentation of the Sensor Network

The segmentation procedure divides the network of sensor points into groups in three stages: first, all the isolated and distant points are removed from the sample. Second, domains containing clusters of "extreme values" are separated from domains of normal values. Third, each domain is segmented into clusters based on the local sampling density values.

1.1.1 Stage 1: Filtering of isolated and distant points

This step ensures that geostatistical analysis excludes values at remote locations which are not correlated with other sampling points. A rectangular box centered at the network's centroid is defined. The extent of the box in the directions x and y is set to $\pm 4\sigma_x$ and $\pm 4\sigma_y$ where σ_x, σ_y are the standard deviations of the sample's coordinate locations. Points outside the boundary box that do not possess a neighbor within a circle of radius equal to $\min(\sigma_x, \sigma_y)$ are removed. For the GDR data in our case study, this procedure removes sensors on remote island locations, e.g., at the Azores or former European colonies, as shown in Fig. 4.1. If necessary, the above step can be applied iteratively.

1.1.2 Stage 2: Partition in "normal-value" and "extreme-value" domains

The GDR data include values generated by two different fields, namely background radioactivity and a spreading plume. The study area is split into two mutually exclusive and jointly exhaustive domains, based on the *threshold value* $x_c = 250$ nSv/h. The threshold follows from an expected range of $0.04 - 0.24$ μSv/h (microSievert per hour) for natural background radiation in Germany [SB07]. The domain of extreme values, henceforth called G2, includes the sensors whose values exceed x_c. The "extreme-value" domain

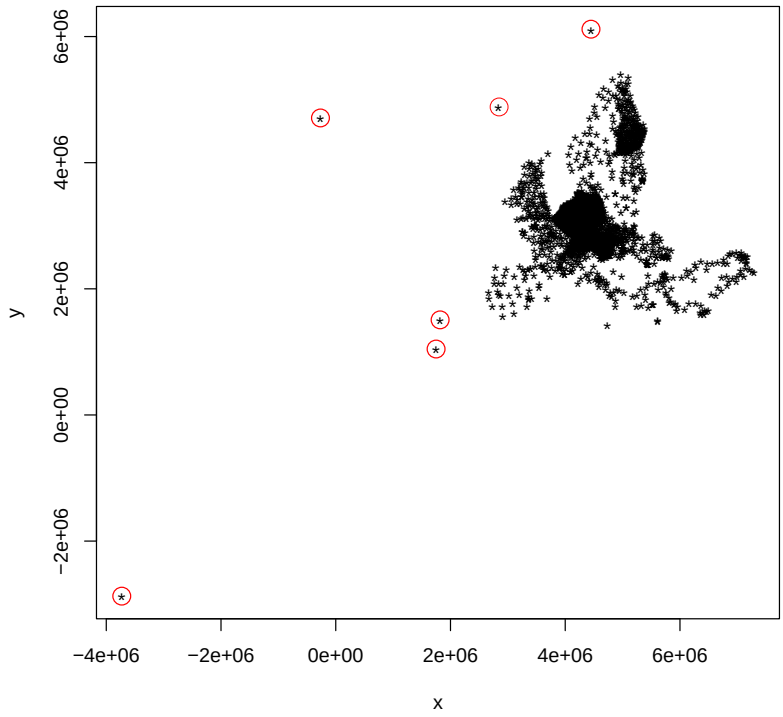


Figure 4.1: Spatial distribution of the sensor grid network over Europe. The distant and isolated points, as identified by the filtering algorithm, are marked by circles.

contains the locations affected by the spreading *radioactive plume*. The remaining sensor points belong to a domain of "normal" values, henceforth referred to as G1. The domain G1 involves points registering *background radioactivity* levels, as well as points with instrument malfunctions, and spikes generated by lightning (certain such events may also be included in G2). The domain G1 contains around 2500 points and G2 approximately 1100 points.

In Fig. 4.2, we illustrate the partitioning on the sensor network into domains of extreme values (black dots, near the center of the network) and normal values (red dots). A map (interpolation) grid of 8000 nodes is also shown (crosses denote the centers of grid cells). Gray (lighter) cells are assigned to G1, while blue (darker) cells are assigned to G2. The assignment is based on the domain identity of the nearest-neighbor sensor to each grid cell. In Fig. 4.3 we plot the natural logarithms of the GDR values registered by the sensors.

This stage can be modified depending on the case study. For example, if one expects that the data follow the Gaussian distribution, spikes due to instrument malfunctions can be filtered using the iterative algorithm proposed in [HMAB07]. Also, different modeling approaches can be used to separate the background from the extreme values (e.g., a trend function could be used to model the spreading plume).

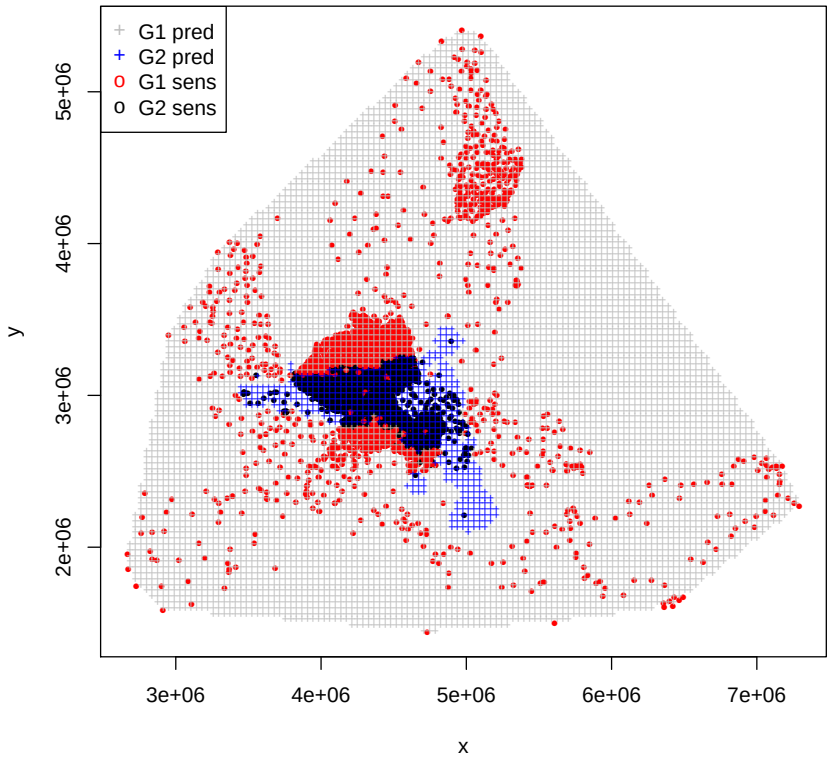


Figure 4.2: Partitioning of the truncated sensor network between the "normal-value" (G1, red dots) and "extreme-value" (G2, black dots) sets. Partitioning of the map grid between the "normal-value" (G1, grey crosses) and "extreme-value" (G2,blue crosses) domains.

1.1.3 Stage 3: Domain partitioning into SSD clusters

In the third stage, we define clusters according to the geographical location and the local sampling density. Clustering based solely on the coordinates

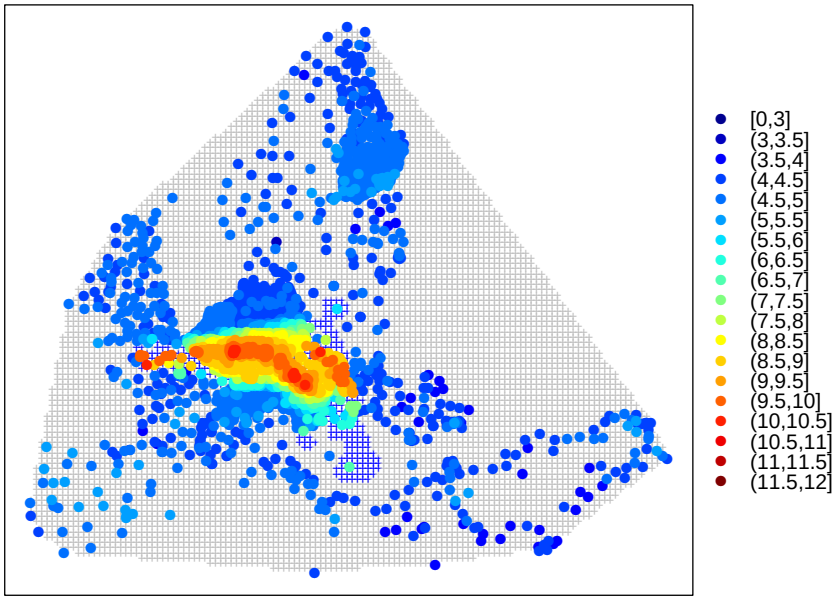


Figure 4.3: Natural logarithms of GDR values at the sensor locations overlaid on the map grid. Grid nodes are partitioned into "normal-value" (G1, gray crosses) and "extreme-value" (G2, blue crosses) domains.

(x_i, y_i) can be performed using various standard methods. Such choices include the mixture of Gaussians [MP00], which is based on the probability density functions of the x_i and y_i; k-means [] clusters the points according to the distances of x_i and y_i from (iteratively defined) cluster centers; support vector machines [CST00]; and the k-median method [GMW07].

However, these approaches are not adequate for our purposes. Given that *CHI* anisotropy estimates are more accurate in areas of higher sampling density, the clustering should aim to exploit the increased accuracy of densely sampled areas. In addition, the sensor network considered here is dynamic, since different sensors may report at different times. It is thus necessary to cluster the sensors without knowing *a priori* the number of clusters.

Hence, we propose the following four-step procedure:

Steps of clustering procedure

1. We construct a sampling density function for the sensor network.
2. We use edge detectors to identify potential clusters as the linked perimeters formed by the edges of the sampling density function.
3. We perform an initial assignment during which sensors are allocated to the linked perimeters that contain them.
4. We reject clusters that include very few sensors. We assign each "orphan" sensor location to the cluster which contains its nearest neighbor (assigned in the initial stage).

The proposed clustering is not motivated by physical reasons and does not imply different anisotropy in each cluster.

Sampling density function To implement the first step, we define a *sampling density grid* (SDG), which is in general different than the map (interpolation) grid. The SDG has $L = \lfloor \sqrt{N} \rfloor$ nodes per side and contains L^2 rectangular cells that cover the sensor network area. For the data set studied, $L = 50$ for the SDG in G1. We construct a sampling density matrix (SDM), which takes at each grid cell a value proportional to the number of sensor points enclosed by the cell. Each sensor point is assigned the sampling density value of the corresponding SDG cell. Algorithms based on density functions [CFZ99, KKK04, AGGR05] are widespread in spatial clustering applications.

Edge detection The second step involves the application of edge detection techniques to determine potential cluster perimeters based on the spatial variability of the SDM [GW06, Chap. 7]. The SDM is first smoothed by an averaging 3×3 filter, to reduce the sensitivity of edge detection to noise. The spatial variation of the smoothed SDM over G1 is shown in Fig. 4.4. The gap between the two main peaks near the center is due to the missing points that belong to G2.

We use the Laplacian of Gaussian (LoG) edge detector, which highlights regions of rapid change and identifies "islands" of similar density [HS92]. The sensitivity of the method is controlled by the edge detector parameters. We opt for a 5×5 LoG filter.

After identifying the candidate "edge" cells, potential cluster perimeters are defined as sequences of linked edge cells. A cell is considered as "linked" if it possesses a neighbor inside a 3×3 neighborhood. The results obtained by applying this procedure are shown in Fig. 4.5.

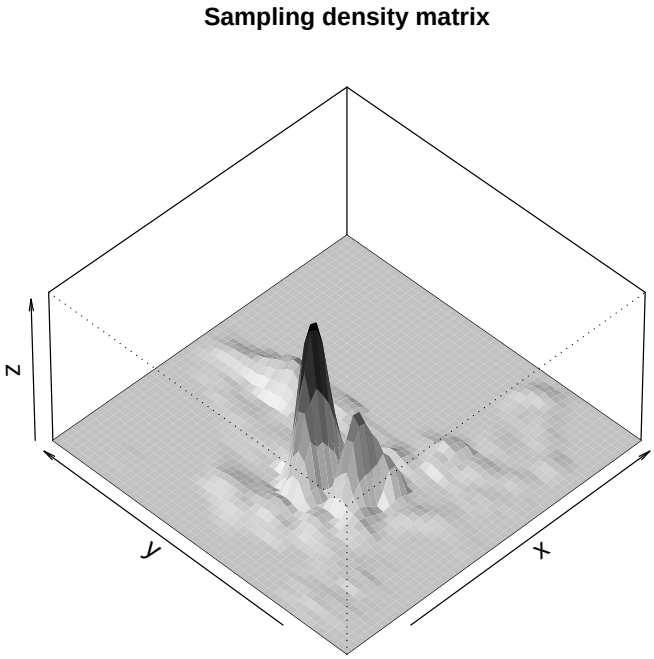


Figure 4.4: Map of smoothed sampling density matrix (SDM) over the background domain G1.

Initial SSD cluster identification After all cells have been searched, each linked perimeter is labeled as a potential cluster perimeter; cf. Fig. 4.5. Sensor points are then assigned to the cluster perimeter that contains them, leading to an initial cluster assignment. Sensor locations that lie outside linked perimeters are not assigned to clusters at this stage. The initial assignment of sampling points is shown in Fig. 4.6. The cluster perimeters are defined using cells of the SDG, but the sampling sites do not in general coincide with the nodes of SDG.

Final SSD cluster assignment Based on our experience, meaningful SSD clusters for *CHI* anisotropy detection should contain at least 25 sensor points. Hence, smaller clusters are rejected. The sampling points inside such clusters as well as unassigned sensor points form the set of *"orphan" points*. The remaining clusters form the set of *admissible clusters*. The orphan points

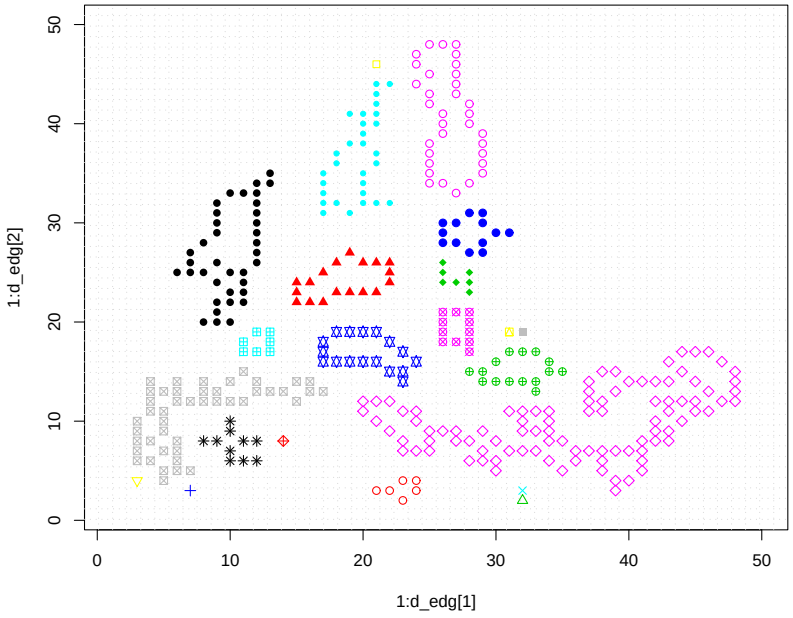


Figure 4.5: Identification of 22 potential cluster perimeters defined on the SDG for the "normal-value" domain (G1). Color markers denote the locations of SDG cells that lie on the identified perimeters. The potential perimeters involves 14 "large" contours and 8 isolated cells.

are assigned to the admitted cluster which contains their nearest assigned neighbor.

All sensor sites are finally assigned to an SSD cluster that includes more than 25 sensor points, as shown in Fig. 4.7. The convex hulls of the seven final clusters are shown in Fig. 4.7. The red dots inside the convex hulls represent the centroids of the final clusters. We use the convex hull for two reasons: (1) For interpolation purposes, prediction points inside the hulls are ensured to have neighbors in the sampling set. This is the approach also used in the Matlab® `griddata` function. (2) The surface areas enclosed by the convex hulls of the clusters are used to obtain relative cluster weights used to derive aggregated anisotropy estimates (see Section 3.2.2).

1.2 Anisotropy Estimation

In the following, we describe the estimation of anisotropy in each SSD cluster based on the *CHI* method. We also propose an expression for deriving

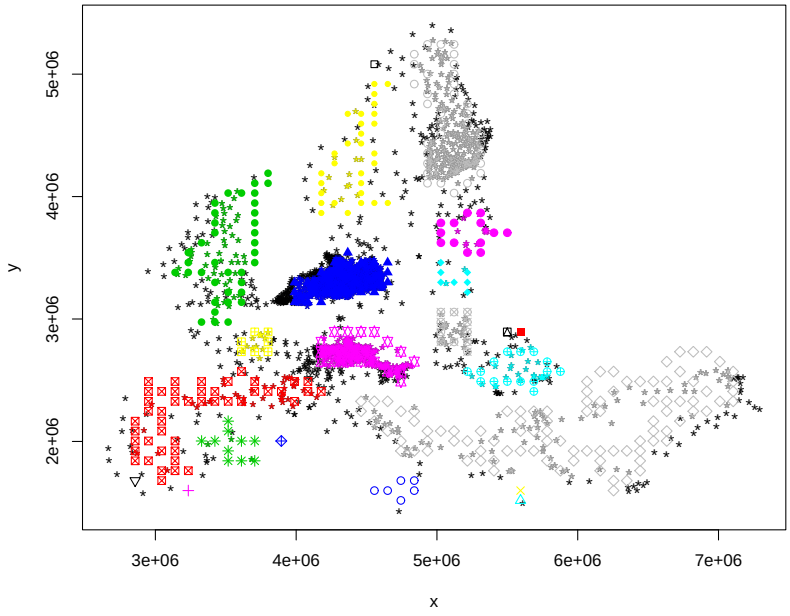


Figure 4.6: Initial assignment of the sampling sites (stars) inside the "normal-value" domain G1 to cluster perimeters. The centers of SDG perimeter cells are denoted by color markers. Colored sampling points are assigned at this stage to clusters, while points marked by black stars are unassigned.

coarse-grained domain averages of anisotropy. Finally, we discuss how the anisotropy estimates are further used in the geostatistical analysis.

1.2.1 Cluster estimates of anisotropy

Estimates of the anisotropy parameters (R, θ) in each cluster are derived using the *CHI* method [CH08]. The angle θ represents the angle between one of the principal axes of anisotropy, arbitrarily called M_1, and the horizontal axis of the coordinate system. In geography, it is preferable to define the anisotropy orientation in terms of the complementary angle. The ratio $R = \xi_1/\xi_2$ is equal to the correlation length along M_1 divided by the correlation length along the orthogonal axis M_2.

Let us define the *gradient tensor* $X_{i,j}(\mathbf{s}) = \frac{\partial X(\mathbf{s})}{\partial s_i} \frac{\partial X(\mathbf{s})}{\partial s_j}$, for $i, j = 1, 2$. If $E[\cdot]$ denotes the ensemble average and $C_{\mathrm{x}}(\mathbf{s}_1, \mathbf{s}_2)$ the *covariance function* of the random field $X(\mathbf{s})$, the *Covariance Hessian Identity* [Swe62] connects the mean gradient tensor to the covariance as follows:

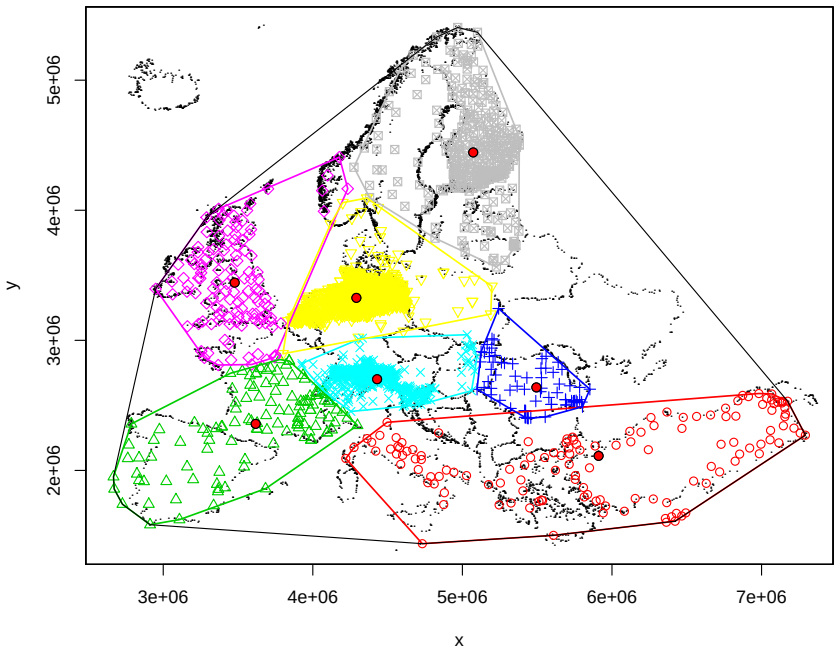


Figure 4.7: Final assignment of sampling points in G1 to seven SSD clusters. Color coding is used for both the sensor points (various markers) and the convex hull boundaries of the clusters. The centroids of the final clusters are marked by red dots.

$$E\left[\frac{\partial X(\mathbf{s})}{\partial s_i}\frac{\partial X(\mathbf{s})}{\partial s_j}\right] = -\left.\frac{\partial^2 C_{\mathrm{x}}(\mathbf{r})}{\partial r_i \partial r_j}\right|_{r=0}. \tag{4.1}$$

Equation (4.1) assumes second-order stationarity and differentiability of $X(\mathbf{s})$ in the mean square sense.

Let us assume that $X(\mathbf{s})$ is second-order stationary with differentiable sample functions, and that ergodic conditions hold (i.e., if the ensemble average of the gradient tensor can be estimated from the sample average). Let $\hat{Q}_{ij}$ represent sample-based estimates of $E[X_{i,j}(\mathbf{s})]$ and $q_{\mathrm{diag}} = \frac{\hat{Q}_{22}}{\hat{Q}_{11}}$, $q_{\mathrm{off}} = \frac{\hat{Q}_{12}}{\hat{Q}_{11}}$ define the diagonal and off-diagonal ratios, respectively. Then, $\hat{R}$ and $\hat{\theta}$ are given by [CH08]:

$$\hat{\theta} = \frac{1}{2}\tan^{-1}\left(\frac{2q_{\mathrm{off}}}{1-q_{\mathrm{diag}}}\right), \quad \hat{R}^2 = 1 + \frac{1-q_{\mathrm{diag}}}{q_{\mathrm{diag}} - (1+q_{\mathrm{diag}})\cos^2\hat{\theta}}. \tag{4.2}$$

If $X(\mathbf{s})$ is Gaussian or log-Gaussian, the existence of the second-order partial derivatives of $C_{\mathrm{x}}(\mathbf{r})$ at zero lag in practice suffices to ensure the differentiability of the sample functions. For a mathematical treatment of sample continuity and its application to partial derivatives of sample functions see [Adl81, p. 63]; for a more applied viewpoint see [Abr97, p. 19-25], and for explicit calculation of sufficient differentiability conditions see [HE07].

In each SSD cluster, the q_{diag} and q_{off} are estimated by means of finite differences on the *rectangular anisotropy estimation grid* that covers the cluster domain. The grid extends from $x_{c;\min}$ to $x_{c;\max}$ in the $x-$direction and from $y_{c;\min}$ to $y_{c;\max}$ in the $y-$direction, where $x_{c;\min} = \min(x_1, \ldots, x_{N_c})$, $x_{\max} = \max(x_1, \ldots, x_{N_c})$, are respectively the smallest and largest values of the x coordinates for all points in the cluster c (similarly for y). To avoid bias related to the cell shape, square grid cells are used with step equal to

$$\alpha_c = \min(|x_{c;\min} - x_{c;\max}|, |y_{c;\min} - y_{c;\max}|)/\sqrt{N_c},$$

where N_c is the number of sensor locations inside the cluster c.

The field values on the anisotropy estimation grids are obtained using a non-parametric, deterministic interpolation approach, such as triangle-based linear or minimum curvature interpolation. This introduces some bias in the anisotropy estimation, since the interpolation model does not account for anisotropy. However, the field generated on the anisotropy estimation grid incorporates the anisotropic properties imparted by the data. The impact of the interpolation method on the anisotropy estimates is studied in [CH08]. In general, dense sampling increases the estimation accuracy of q_{diag} and q_{off}.

1.2.2 Coarse-grained domain estimates of anisotropy

Interpolating $X(\mathbf{s})$ on a map grid using cluster estimates of anisotropy would require a smoothing filter, e.g., moving windows. Alternatively, one can seek an average estimate of anisotropy (over the clusters). Since we consider domains with different statistical properties, the average should be conducted separately in each domain. Given the nonlinearity of the *CHI* expressions in (4.2), a simple average of the cluster anisotropy parameters is not appropriate.

Let us assume that each domain involves K_g clusters ($g = 1, 2$ in our case), and that $\hat{Q}_{ij}^{g;c}$, $i, j = 1, 2$, $c = 1, \ldots, K_g$ represent the estimates of the mean gradient tensor for the c-th cluster in the g-th domain. Anisotropy estimates are based on the *weighted average*, $\overline{Q_{ij}^g}$, of the cluster gradient tensors:

$$\overline{Q_{ij}^g} = \frac{\sum_{c=1}^{K_g} w_{g;c}\, \hat{Q}_{ij}^{g;c}}{\sum_{c=1}^{K_g} w_{g;c}} \tag{4.3}$$

The weights $w_{g;c}$ are set equal to the area $A_{g;c}$ enclosed by the convex hull of each cluster. The values $\overline{Q_{ij}^g}$ are then used in Eqs. (4.2).

1.3 Incorporating CHI Anisotropy Estimates in the Variogram Model

If the inference of the spatial model employs the experimental variogram, $(\hat{R}, \hat{\theta})$ can be used to rotate and rescale the coordinate system to render the spatial dependence isotropic [Hri02, CH08]. Then, the omnidirectional empirical variogram can be estimated and modeled. Spatial interpolation is performed in the transformed coordinate system using the optimal isotropic variogram model. In this case the transformed values of the map grid coordinates should be used. Alternatively, $(\hat{R}, \hat{\theta})$ can be used as educated initial guesses of the anisotropy parameters in maximum likelihood optimization. This approach can lead to more accurate estimates of anisotropy, but it is computationally prohibitive for large data sets.

The anisotropy parameter estimates given by (4.2) are sample statistics and thus exhibit sample-to-sample fluctuations: a realization of an isotropic random field may have $\hat{R} \neq 1$. Hence, we need a statistical test for the hypothesis that a data set is isotropic at a specified significance level. If the isotropic hypothesis cannot be rejected, isotropy restoring transformations can be skipped to reduce the computing time, without significant impact on the accuracy of interpolation.

A non-parametric joint probability density function has been developed and its confidence regions have been calculated [PH10]. These can be used to test whether the isotropy assumption can be rejected at a given confidence level. More specifically, the isotropy hypothesis cannot be rejected if

$$\hat{R}^2 \in \left(\frac{N_c - 2\sqrt{(N_c - r_\alpha) r_\alpha}}{N_c - 2r_\alpha}, \frac{N_c + 2\sqrt{(N_c - r_\alpha) r_\alpha}}{N_c - 2r_\alpha} \right), \tag{4.4}$$

where r_α is a constant that depends on the desired confidence level (for 95% confidence $r_\alpha \simeq 6$). In (4.4) $N_c \geq 25$ is the number of sampling points involved in the estimates: for a single domain and a single cluster $N_c = N$, while for a single domain with multiple clusters $\sum_{c=1}^{K} N_c = N$. The test is conservative (as shown by theoretical arguments and numerical simulations), i.e., it overestimates the width of the confidence region due to underestimation of correlation effects. The accuracy of the test is compromised for small data sets or sparsely sampled areas, due to poor estimation of $(\hat{R}, \hat{\theta})$.

In practice, statistical significance does not directly translate into interpolation performance. If $N_c \gg 1$, the isotropic confidence interval of $\hat{R}$ is

very narrow; hence, a small deviation of $\hat{R}$ from 1 may imply statistically significant anisotropy, while it has little or no effect on the interpolation due to the abundance of data. In this case we incur a relatively small computational cost by performing an isotropy restoring transformation for little or no gain in performance. Reversely, if N_c is small, the test may show that a larger deviation of $\hat{R}$ from 1, potentially significant in the interpolation, may be statistically insignificant, simply because there are not enough observations. In this case, one may think that an isotropy restoring transformation is a good idea, even though the isotropy test does not call for it. However, we should keep in mind that the estimate of $\hat{R}$ is also affected by the data sparseness, and thus its value is highly uncertain. Essentially, if N_c is small the ability to accurately resolve the anisotropy deteriorates significantly.

Chapter 5

Anisotropy Analysis of Synthetic Data

In this chapter we will present synthetic data-sets used for the performance analysis of *CHI* and *CCHI* algorithms. The synthetic scenarios are realizations of Gaussian Random Fields (GRF) generated via the Fast Fourier Transform methodology described later in subsection 1. In addition, some of the on-grid GRFs are sub-sampled in order to generate scatter data-sets. Several realizations of these scattered data-sets are combined in order to generate clusters with various anisotropy parameters as presented in subsection 2. Synthetic scattered sensor networks are used to validate *CCHI* algorithm performance.

1 Construction of Synthetic Gaussian Random fields using the FFT spectral method

Several methods have been proposed for the generation of random fields both on regular and irregular grids. A very popular one uses Fast Fourier Transform as it is extremely beneficial if the field is sampled over a regular (rectangular) lattice. Below we present the basic steps of the method.

1. Construction of a lattice in real space for covariance functions that admit explicit real space expressions. It is also possible to construct an equivalent lattice in spectral space, which is useful for covariance functions that are valid only in spectral space such as the Spartan Covariance Functions presented in [Hri03].

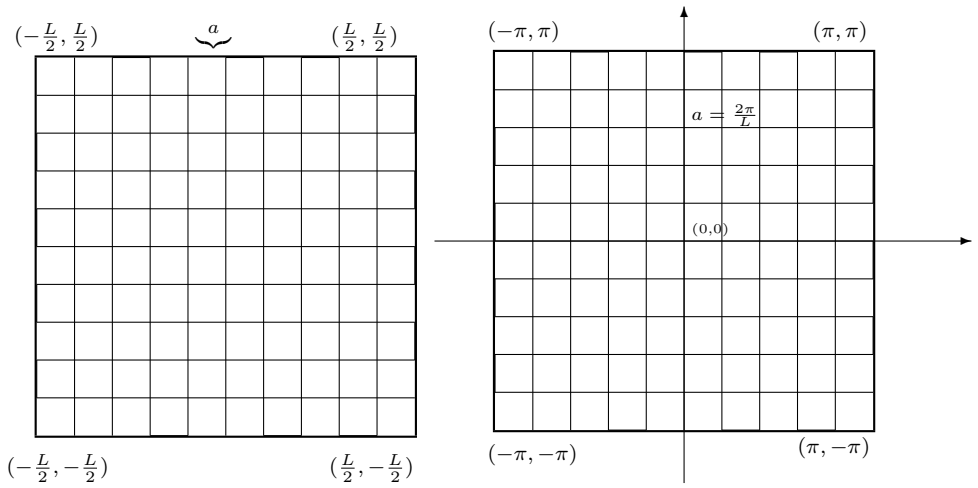


Figure 5.1: Construction of a lattice in real space (left) and in spectral space (right).

2. Evaluation of Euclidean distances off all lattice points on real from the reference point $(0, 0)$. Each coordinate is scaled with the respected correlation length (ξ_i, ξ_j) and rotated θ degrees to include anisotropy in the Random Field. The distance matrix H is calculated according to the new coordinates. The H given in 2-D from

$$\begin{bmatrix} x/\xi_x \\ y/\xi_y \end{bmatrix} \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

$$H(i,j) = \sqrt{x'(i,j)^2 + y'(i,j)^2}$$

3. Evaluation of the desired covariance function on each point of the lattice using the distance matrix H.

Exponential	$\sigma^2 \exp(-\lvert H \rvert)$
Gaussian	$\sigma^2 \exp(-\lvert H \rvert^2)$
Spherical	$\sigma^2(1 - 1.5H + 0.5H^3)$, $0 < H < 1$, and 0 otherwise
Matérn	$\sigma^2 \frac{1}{\Gamma(\nu)2^{\nu-1}} (2\sqrt{\nu}\frac{d}{\rho})^\nu K_\nu (2\sqrt{\nu}\frac{d}{\rho})$

4. Construction of a random the fluctuation matrix using a real number random matrix and calculating its Fast Fourier Transform. This way we ensure that the inverse FFT will yield real numbers.

$$U = \texttt{fft}(randn([LL]))$$

where L is the number of cells per direction.

5. Multiplication point to point of the random matrix with the square root of the spectral covariance function $\widetilde{C}$ and normalization of the random fields according to lattice size.

$$\widetilde{X}_f = \sqrt{\widetilde{C}}U/L^2$$

6. The inverse Fast Fourier Transform projects the constructed field back to real space coordinates. Finally, we add the mean value of the simulated field μ.

$$X = \texttt{ifft}(\widetilde{X}_f) + \mu$$

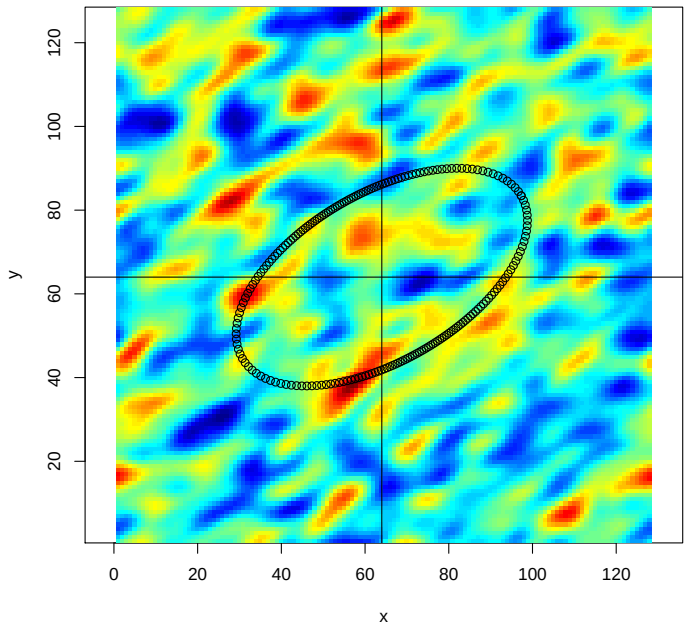


Figure 5.2: Synthetic Gaussian Random field N(100,10) on a 128-by-128 square lattice, constructed with the `FFT` method. Although the ellipse semi-axes lengths are not shown in scale, the ellipse visualizes the geometric anisotropy as estimated by *CHI* by means of $(\hat{R}, \hat{\theta})$. The correlation lengths of the constructed fields along 30 and 120 degrees are 4 and 8 respectively.

2 Construction of a synthetic sensors network from GRF

In order to examine the performance of our clustering algorithm and evaluate the results of the combination of Clustering and *CHI* method, we generated artificial sensor networks. The main idea is to sample each cluster from a different GRF realization and then to project all the samples from different clusters to a common coordinate system. Each GRF realization is generated on lattice and sampled later on. The mean value, the anisotropy parameters, the node's length, and the selected samples change at each realization. On the other hand centroids of the clusters are fixed.

An example of the datasets produced through this procedure is demonstrated in figure (5.3). Visual inspection yields the existence of 4 different clusters. All clusters were produced over a 32×32 grid using different anisotropy parameters (see legend in figure 5.3).

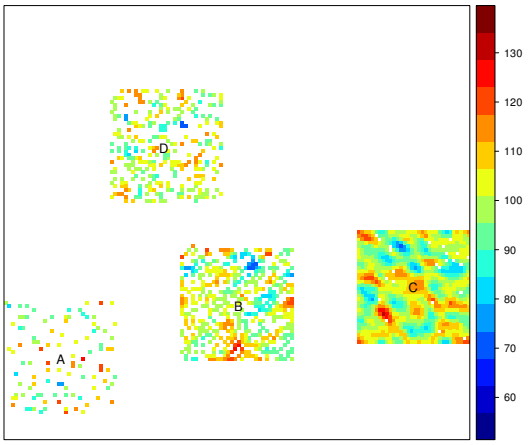


Figure 5.3: A realization of the artificial sensor network. Clusters A,B and D are sampled from a GRF 32×32 generated on a square grid with anisotropy parameters $[R = 2, \theta = 30°]$. The sample size for these clusters are 100,500 and 300 respectively for each cluster. Cluster C includes 1000 points sampled from an equal size GRF of with parameters $[R = 1.5, \theta = -30°]$.

3 Performance of *CHI*

In [Hri02, CH08], it was shown that the *CHI* method provides accurate estimates of anisotropy parameters of two-dimensional normal and log-normal on-grid random fields. In addition the *CHI* method was applied to irregular samples using bi-linear, bi-cubic and biharmonics spline interpolation methods to project the information on-grid. As expected, the accuracy of the estimates of irregular samples (highly) depends on the choice of the interpolation method and the spatial distribution of the data. This study examines how the *CHI* estimates depend on sampling density and spatial distribution of the data.

3.1 The role of sampling Density

In order to examine how the sampling density of the data affects the anisotropy estimation, we took the following steps. First, we constructed anisotropic Gaussian random fields on a square grid (see also sub-section 1). Second, we used different degrees of random sub-sampling to generate synthetic irregular samples that follow the original probability distribution (Fig 5.4, 5.5). Third, we constructed a new grid whose number of cells is equal to the number samples as proposed in [CH08]. Finally, we chose bi-linear interpolation to estimate the values of the fields on the interpolation grid in order to apply the *CHI* method. The choice of bi-linear interpolation meets goals of both performance and speed. Since bi-linear interpolation is much faster than biharmonics splines, while the accuracy of estimates is not compromised.

We generated randomly 200 GRF with arbitrary R and θ in ranges of $[1, 3]$ and $[-45, 45]$ respectively. Figure 5.6 shows the bias of anisotropy estimates R and θ in respect to the percentage of pixels in the sub-sampled realization. Notice that the accuracy of the estimates is proportional to the sample size. As expected, the gaps among sample points decrease the accuracy of the *CHI* method.

$\hat{R}_{sub}$	$\hat{\theta}_{sub}$	$\hat{R}_{full}$	$\hat{\theta}_{full}$
1.16	-9.71°	1.19	-8.82°

Table 5.1: The estimates of the image for the full set $(\hat{R}_{full}, \hat{\theta}_{full})$ and the selected subset $(\hat{R}_{sub}, \hat{\theta}_{sub})$ presented in figure (5.4).

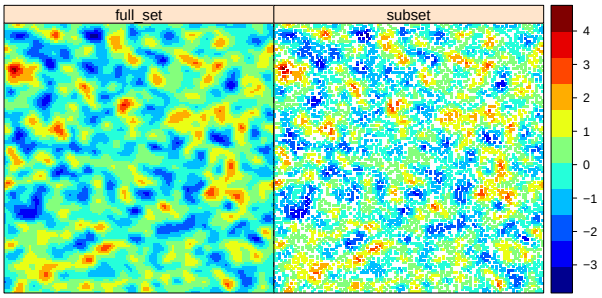


Figure 5.4: Left: an original (full) realization of a GRF N(0,1) on grid 128x128 square grid. Right: the selected subset (sub).

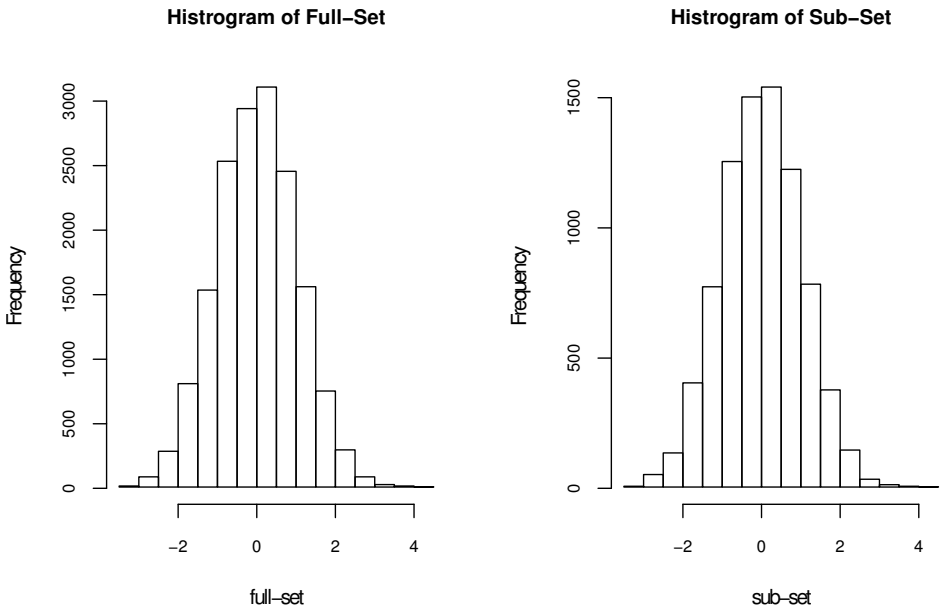


Figure 5.5: Full (left) and subset (right) histograms respectively. The plots show that the subset and the full dataset follow the same distribution.

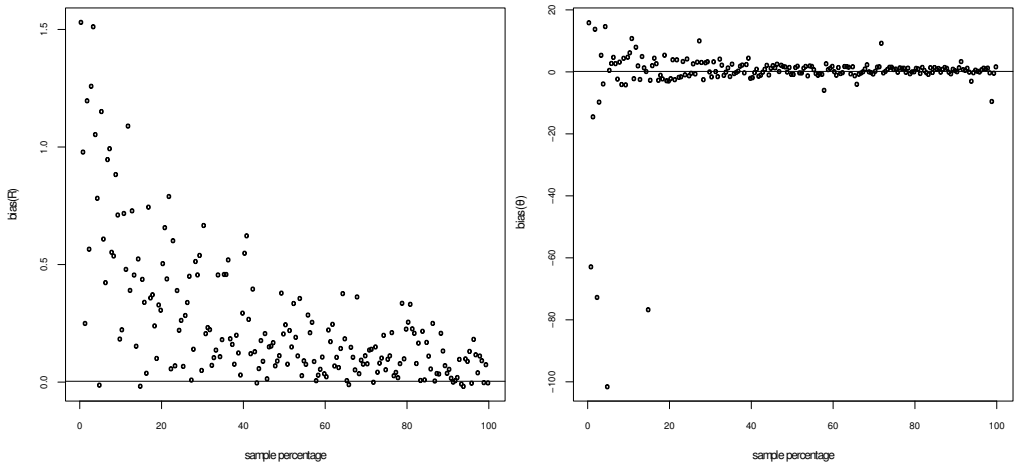
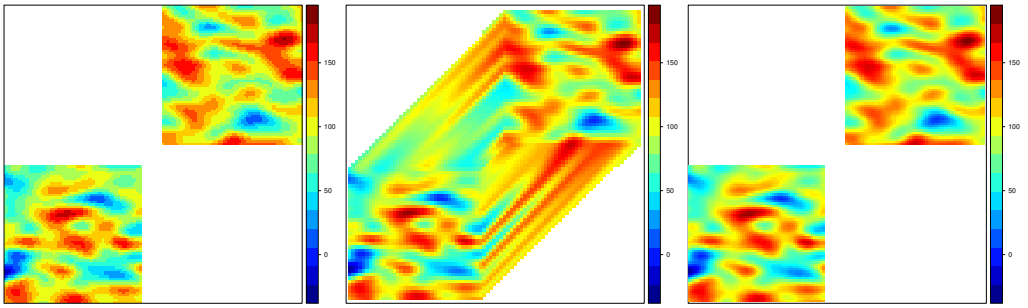


Figure 5.6: Bias for R (left) and θ (right) estimates with respect to the percentage of pixels in the sub-sampled realization.

3.2 The role of dispersion in the sampling sites



(a) (b) (c)

Figure 5.7: (a) The original dataset. (b) Single cluster linear interpolation of the two disjoint areas.(c) Linear interpolation on each area-cluster separately.

We have shown so far that *CHI's* performance depends on the interpolation result of the scattered data. In addition, the interpolation result significantly depends on the dispersion of the data sites. By dispersion we refer to both spatial density and sets formation of the data. If the sampled data contain large empty spaces among them, it is desirable to interpolate each set separately. Figure (5.7) shows how interpolation between two visual clusters may change the anisotropy of the field. The two square clusters on the left have been extracted from the same RF realization on a 128x128 square grid with anisotropy parameters $[R = 2, \theta = 0]$

	CHI	*CCHI*		$cluster_1$	$cluster_2$
$\hat{R}$	1.66	2.08		2.15	2.03
$\hat{\theta}$	12.15°	-4.30°		-1.64°	-8.02°

Table 5.2: Anisotropy estimates for figure 5.7. Columns *CHI* and *CCHI* correspond to anisotropy estimates for subfigures (b) and (c) respectively. Columns $cluster_1$ and $cluster_2$ correspond to the *CHI* anisotropy estimates for the two disjoint areas calculated separately (as shown in subfigure (a)).

In order to determine whether this behavior is consistent for this type of datasets we generated 200 GRF realizations on a 128x128 lattice with anisotropy parameters $R \in [0, 3]$ and $\theta \in [-45, 45]$. Then, we removed about half of the observation points, mostly on the upper left and lower right of the grid, in order to achieve clusters equivalent to figure 5.7's sampling structure.

Table 5.3 shows mean E_{200} and standard deviation σ_{200} of *CHI* and *CCHI* bias over the 200 runs. (R^i, θ^i) are the anisotropy parameters that

the i^{th} realization of the field(the parameters that the realization was created with). $[R^i_{CHI}\,, \theta^i_{CHI}\,]$ and $[R^i_{CCHI}\,, \theta^i_{CCHI}\,]$ are the corresponding estimates of *CHI* and *CCHI* respectively for this realization.

	$E_{200}[R^i - \hat{R}^i_X]$	$E_{200}[\theta^i - \hat{\theta}^i_X]$	$\sigma_{200}[R^i - \hat{R}^i_X]$	$\sigma_{200}[\theta^i - \hat{\theta}^i_X]$
CHI	0.33	-17.58°	0.54	17.83°
CCHI	0.07	1.79°	0.15	7.01°

Table 5.3: Mean bias (E_{200}) and standard deviation of the bias (σ_{200}) based on 200 runs for $R^i \in [0, 3]$ & $\theta^i \in [-45°, 45°]$. The dispersion of the datasets is shown above in Fig. 5.7.

One can notice, that *CCHI* outperforms *CHI* . This result is expected since linear interpolation fills the gap between clusters with a completely different pattern. To conclude, on the one hand there is strong evidence that clustering significantly improves anisotropy estimates in cases where the spatial support of the data is fragmented. On the other hand, if a dense cluster is split there is no significant loss of accuracy.

4 Performance of *CCHI*

In order to examine the impact of different clustering algorithms on anisotropy estimation we generated synthetic clusters. Each sample point is assigned to a certain cluster based on an index number. Only samples points that belong to the same cluster are sampled from the same realization of the field. Each cluster corresponds to a different realization for a single validation run. Moreover on each validation run all the cluster realizations change.

In addition we selected a fixed the number of clusters (equal to four) and a fixed maximum number of points per cluster (equal to 200,400,800, and 2000 for each cluster). However the subsample percentage is random for each cluster. The parameters of the each realization are also chosen randomly within range shown later in table 5.4. Finally, the parameter *sp* controls the size of the generation lattice's cell and defines the dispersion of the samples for each cluster.

Each GRF realization was generated following the same methodology presented in section 2 of this chapter. The samples spatial distribution aims to simulated a realistic multi-country monitoring sensor network.

Evaluation criterion is the bias between the estimated and original anisotropy parameters from the entire domain. Given the original index, i.e. the index that assigns all samples from the same realization to the same cluster, we calculate a pair of anisotropy parameters for the entire domain as presented in subsection 4.1.2.2. In the same way we estimate a pair of anisotropy parameters based on index that each clustering algorithm calculates.

Number of cluster per run	$N_c = 4$
Maximum number of sensors in clusters [a,b,c,d]	$N \in [200, 400, 800 \ \& \ 2000]$
Mean value on each cluster	$m \in [80, 400]$
Anisotropy ratio on each cluster	$R \in [1, 3]$
Anisotropy direction on each cluster	$\theta \in (-45°, 45°)$
Spreading of each cluster on common coordinate system	$sp \in [0.5, 1.5]$

Table 5.4: Statistics per cluster in synthetic data. Note that : 1) Upper limit of points is different for each cluster. 2) Spreading affects the density of the cluster

Figure 5.8 shows a realization of the simulated scenarios. Figure 5.9 shows the clustering results of the *CCHI,* DBSCAN, 4-means and x-means algorithms on the same run. We cannot claim that this is a completely fair comparison of these algorithms. Except for x-means and *CCHI,* the other two algorithms have no automatic procedure for estimating the number of clusters. For this reason, we fixed $k = 4$ for k-means and in addition $eps = 0.04$

and $m = 6$ for DBSCAN (*eps* denotes the neighborhood relative to the entire domain and m is the number of points inside *eps*).

The average error of (R, θ) and its standard deviation for 100 runs of each algorithm are shown in table 5.5. Notice that *CCHI* marginally performs better of the other algorithms. We do not claim that this is the best algorithm for clustering in general. We claim though that it performs equally well to other algorithms, in a large variety of data that contain both dense and sparse areas, with minimal parameters configuration. The choice of 3x3 of 5x5 averaging filter for *CCHI* classifies a very large set of instances. All examples on this thesis were performed using a 3x3 averaging filter.

X=	CCTI	4means	DBSCAN	Xmeans
$E_{100}[R^i - \hat{R}^i_X]$	0.041	0.044	0.048	0.052
$\sigma_{100}[R^i - \hat{R}^i_X]$	0.089	0.090	0.107	0.102
$E_{100}[\theta^i - \hat{\theta}^i_X]$	-0.356°	0.331°	-2.752°	-8.117°
$\sigma_{100}[\theta^i - \hat{\theta}^i_X]$	31.082°	29.824°	33.687°	32.892°

Table 5.5: Average and standard deviation (100 runs) of the estimation bias using four different clustering algorithms.

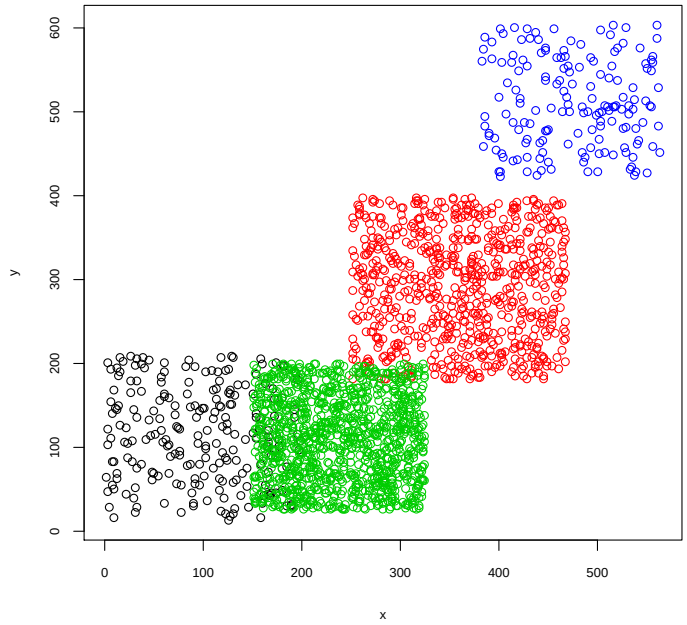
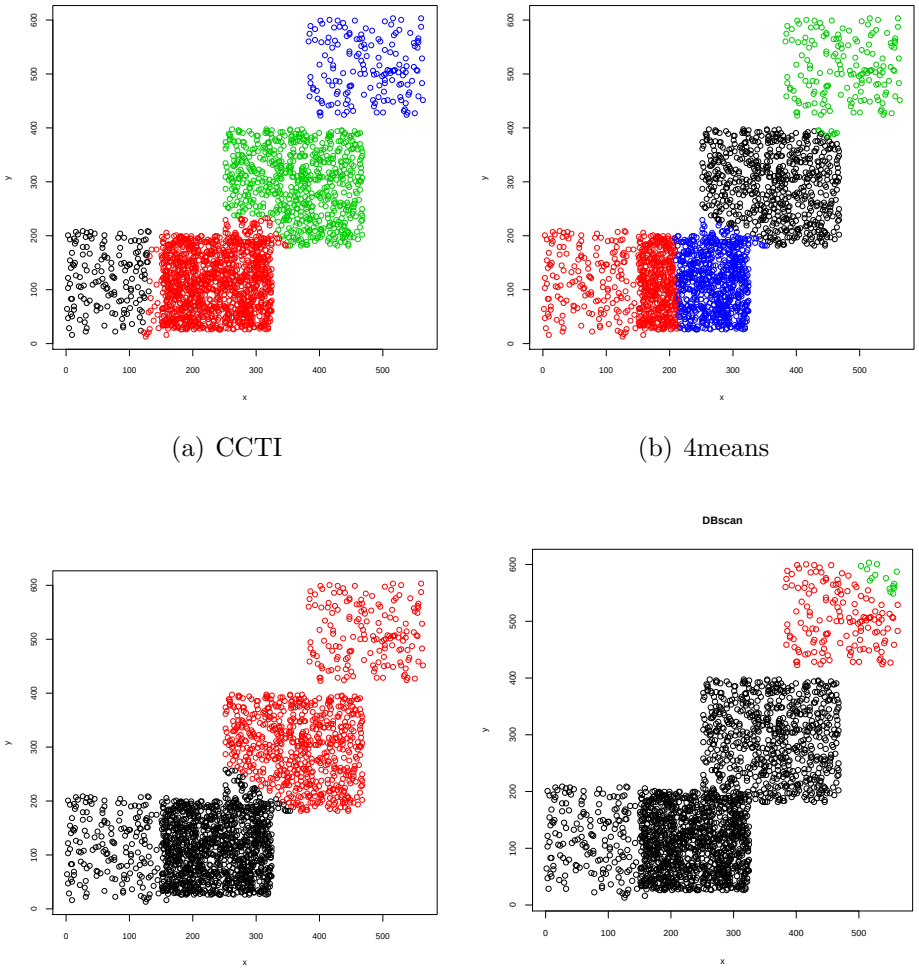


Figure 5.8: Single realization of the synthetic network . Each color represents different clusters and in respect samples selected from a different GRF realization (see table 5.4).



(a) CCTI (b) 4means

(c) xmeans (d) DBSCAN

Figure 5.9: The original index (corresponds to the realization indices) of the dataset used is shown in figure 5.8. **Top**: The clustering result (single run) for the *CCHI* and 4-means methods on (a) and (b) respectively. **Bottom**: Left: X-means result with cluster variable number in the range [2,5]. Right: Results for DBSACN with parameters $eps = 0.04$ and $m = 6$ (range of the neighborhood and smaller cluster size respectively).

Chapter 6

Anisotropy Analysis of Real Data

Two different real-case scenarios provided by the *European Radiological Exchange Platform* (EURDEP) Gamma Dose Rate (GDR) network are described in this chapter. This chapter opens with a presentation of the *rain-event* scenario which monitors the radioactivity over the European continent during episodes of heavy rainfalls. Second, *worst-case* scenario monitors the GDR over Europe several hours after a simulated nuclear accident at a nuclear facility in Belgium. The performance of *CCHI* is compared to other clustering algorithms in terms of anisotropy parameter estimation. In addition, a cross- validation analysis for the accuracy of the worst case scenario is presented here, along with prediction maps generated by means of kriging.

1 Rain-events Scenario

This dataset includes GDR data from the real-time monitoring network of the *European Radiological Exchange Platform* (EURDEP) from 16^{th} to 19^{th} September 2006. During that time period, heavy rainfalls were reported all over central Europe. According to [USB09], rainfall tends to enhance GDR measurements, as the natural activity in the air is washed out. The GDR enhancement occurs for a time period between 30 and 120 minutes maximum. As authors of [USB09] state, due to reporting differences among the sensor networks, the GDR data have been averaged hourly and aggregated with EURDEP daily averages. The final dataset contains measurements in two hour intervals for the three day period; i.e missing time slices are present due to missing data.

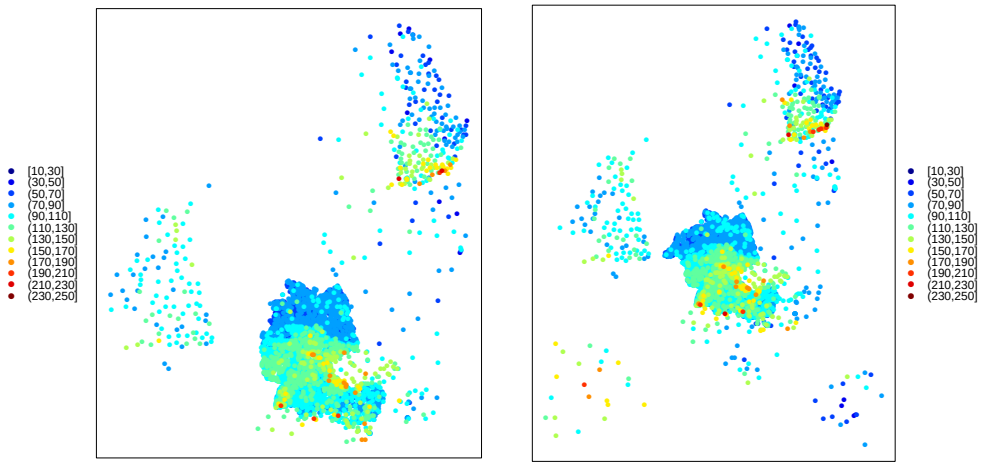


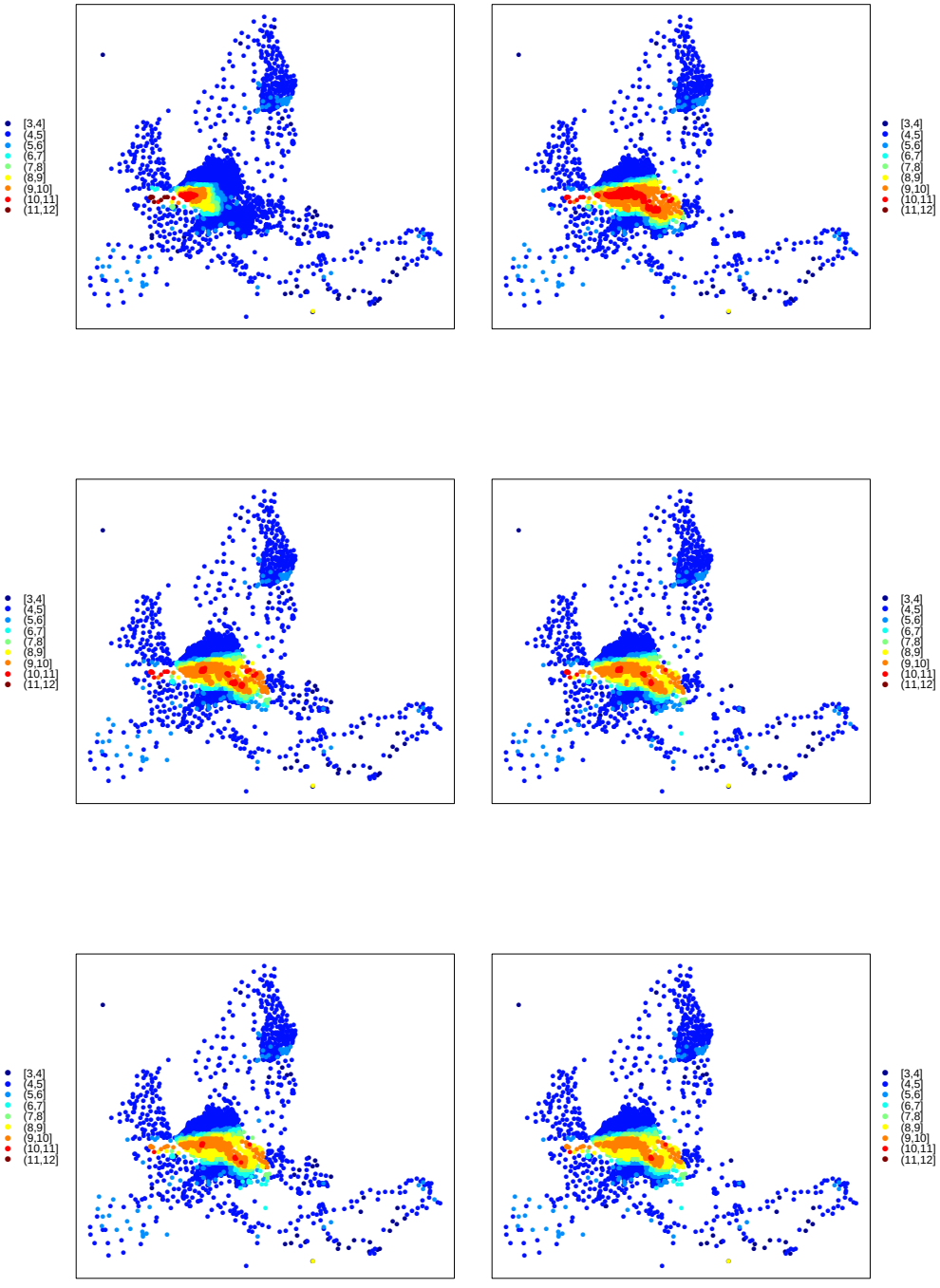
Figure 6.1: Two time instances of Rain-event scenario.

2 Worst-Case Scenario

This scenario involves a total of $N = 3626$ sampling sites with their positions expressed in the INSPIRE coordinate system [INS09]. GDR is measured in *nanoSievert per hour (nSv/h)*. The network involves both densely sampled areas (e.g., Germany and Austria) and sparsely sampled ones (e.g. South Europe).

Real background radioactivity measurements are combined with simulations that include systematic errors, local peaks due to washout effects caused by heavy rainfall, single peaks due to lighting strikes, and areas of extreme values resulting from the dispersion of a radioactive plume caused by a simulated reactor accident in central Europe. The simulations are generated with the RODOS system [Ehr97] using meteorological information from the German weather service. The time of the simulated accident was 23:40 on January 6, 2008. Forecasts of the plume dispersion were produced at +18h, +30h, +42h, and +54h from the starting time, for over an area covering 2500×2500 km^2 centered at the city of Offenbach.

Figure 6.2: The images below present the natural logarithm of the measured GDR values over Europe for the simulated "worst case" scenario. In this simulated scenario, a reactor of a nuclear plant, located in Belgium, explodes and the radioactive plume is transferred throughout the air. We present 6 measurements taken at 12h intervals between them, starting with an 18h delay from the simulated accident.



3 Application to Real Data

In the previous section we showed that minimizing data gaps inside and between clusters increases the accuracy of anisotropy estimation. However, when working with real case scenarios, such as those presented in figures (6.3 & 6.4), we come across a modeling dilemma: On the one hand, one can aim to minimize gaps by leaving out scattered stations. On the other hand, these stations may contain important information that should be included in the calculations. Unfortunately there is no convenient answer to this situation.

The original goal of this work was to be able to provide anisotropy estimates of GDR from a European sensor network in an automatic procedure. Both rain-event and worst-case scenarios present different instances of "difficult situations". As one may notice by examining these different datasets, the network density differs from country to country. In addition, countries do not always broadcast their measurement to the network at the same time, leaving large time and space gaps among live sensors.

Method	$\hat{R}$	$\hat{\theta}$
CCTI	1.05	38.25°
4means	1.04	39.33°
DBSCAN	1.11	-15.97°
Xmeans	1.09	11.01°

Table 6.1: Anisotropy estimates for rain-event scenario

We provide weighted average of anisotropy to compare the clustering algorithms. The weighted average is calculated by means of coarse-grained anisotropy estimates as explained in section 1.2.2. If we examine table 6.1, we may notice that 4-means and *CCHI* estimates are very close, while the remaining estimates seem to have different directions. Although there are differences, all results seem to be quite isotropic, so it is not safe to extract conclusions. However, in figure 6.3, we may notice that the clustering of *CCHI* and 4-means is almost identical, and dividing the Swedish network into two clusters does not affect the result. In view of similar behavior in synthetic data, we point out that splitting a dense cluster does not affect the coarse-grained anisotropy estimates if the anisotropy does not change inside cluster. On the other hand, it has been shown (section 2) that merging two distant clusters may decrease accuracy of anisotropy estimates (see x-means, 6.3,c). Finally, we observe in figure 6.4 that black squares in DBSCAN clustering are considered as noise. These sensors do not meet cluster definition of DBSCAN (see def 7). It is not easy to tell whether these stations improve

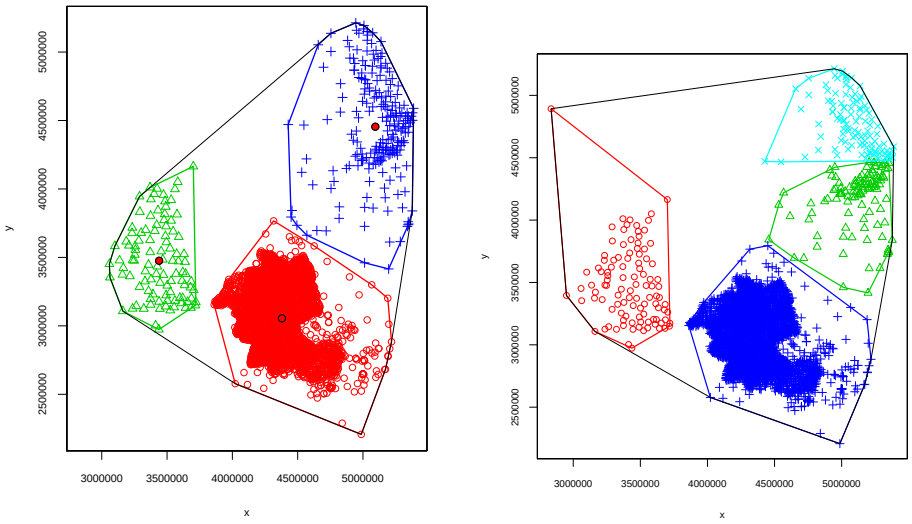
the anisotropy estimates or not.

Method	$\hat{R}$	$\hat{\theta}$
CCTI	1.378	-17.84°
4means	1.50	-16.08°
DBSCAN	1.46	-15.48°
Xmeans	1.46	-16.00°

Table 6.2: Anisotropy estimates for worst-case scenario.

One may notice similar behavior in the worst case scenario (fig 6.4). The k-means (actually 4-means) algorithm does not take into account the sampling density of the data and favors large gaps among data. Moreover x-means, which is a generalization of k-means, splits these data into two clusters, leaving even larger gaps to be interpolated. DBSCAN, is designed though to minimize these gaps. When there are areas that have a dense network, like Central Europe, and others with only a few scattered networks like Eastern Europe, it is hard to define the optimum eps-neighborhood . If the eps-neighborhood is too large, then all points are assigned into a single cluster. If eps-neighborhood is too small, then only dense clusters will be found, leaving a large number of sensors unexploited.

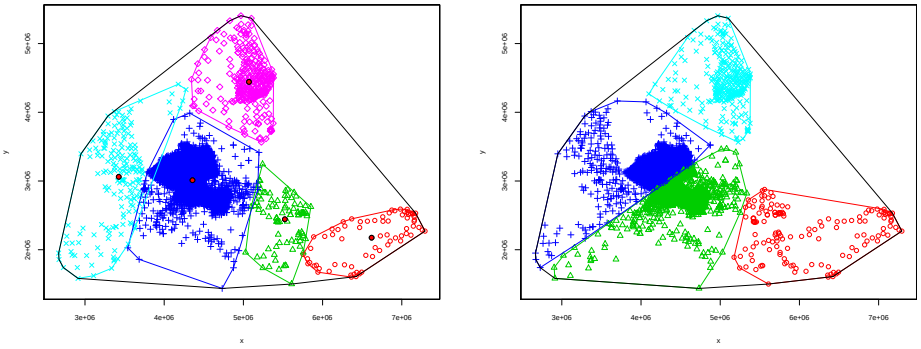
The anisotropy estimates of the worst-case scenario shown in table 6.2 exhibit similar behavior for all four clustering algorithms. In fig 6.2, one notices a highly radioactive plum with West to South-East orientation. The direction of the plume agrees with the anisotropy estimate. The plume values are extremely high compared to the background values. As explained above in section 4, these values form a new stochastic-process and should be treated separately.



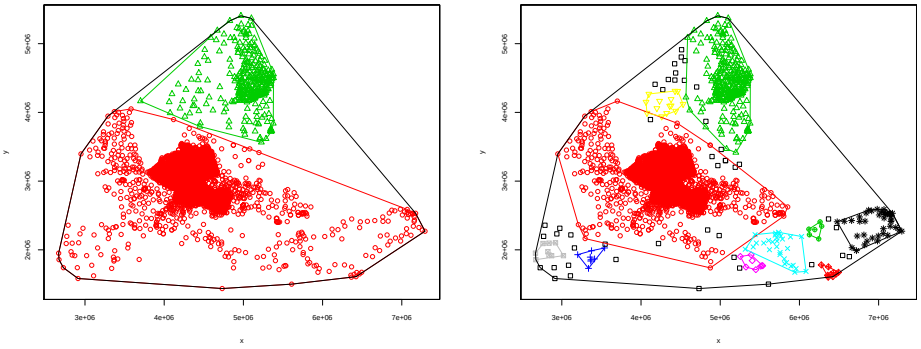
(a) CCTI (b) 4means

(c) xmeans (d) DBSCAN

Figure 6.3: Rain-event Scenario single run



(a) CCTI (b) 4means



(c) xmeans (d) DBSCAN

Figure 6.4: Worst case scenario single run

4 Cross-validation Analysis of Anisotropy Estimates Clustered *CHI*

We used the +42h time slice of the worst case scenario for the spatial analysis. The statistics of the GDR data, given in Table 6.3, exhibit large variability and strong deviations from Gaussianity.

Table 6.3: Statistics of the GDR data used in the case study. The symbols used denote the following: $x_{\min}$ (minimum value), q_1 (first quartile), q_2 (median), m_{x} (mean value) q_3 (third quartile), $x_{\max}$ (maximum), σ_{x} (standard deviation), μ_{x} (skewness), k_{x} (kurtosis).

$x_{\min}$	q_1	q_2	m_{x}	q_3	$x_{\max}$	σ_{z}	μ_{x}	k_{x}
29.0	85.8	131.0	2442.0	3082.0	26990.0	4371.36	2.29	5.25

We have conducted tests on single-domain, single-cluster synthetic and real data (not shown here), which show that application of the *CHI* improves interpolation performance. Here we test the potential benefits of anisotropy estimation for mapping of the radioactivity distribution by a cross-validation approach on the GDR data set.

4.1 Study Design

In the following, we investigate the effect of incorporating anisotropy in the variogram model on the performance of interpolation by ordinary kriging. We also consider the effect of partitioning the study area into domains of normal and extreme values. We evaluate the performance by means of cross-validation analysis and by visual inspection of kriged maps.

To calculate cross-validation measures, we consider different partitions of the $N = 3626$ points into a *training set* that contains $N_t = N - \lfloor p\,N/100 \rfloor$ points and a *validation set* containing $N_v = \lfloor p\,N/100 \rfloor$ points. The GDR values at the validation points are set aside for comparison with the predicted values. For each partition $p = (10, 30, 60, 90)$, fifty (50) *sampling realizations* are generated by randomly selecting the training points, which are replaced at the end of each run. We consider four approaches for spatial interpolation that employ ordinary kriging (OK) to estimate the radioactivity field at the validation sites.

The first approach (abbreviated as ID in Table 6.4) segregates the training set into "normal-value" (G1) and "extreme-value" (G2) domains. The range of values in G1 is $29.0 - 248.8$ nSv/h, while in G2 it is $251.0 - 26992.5$ nSv/h.

A separate variogram is estimated in each domain assuming isotropy. Each validation point is assigned to one domain based on the domain identity of its nearest neighbor in the training set. The predictions at the validation points use the variogram function of the domain of ownership. The second approach (abbr. AD) differs only in the fact that anisotropy parameters are estimated, and their values are used to perform an isotropy restoring transformation before the omni-directional experimental variograms are estimated and fitted in each domain.

The third approach (abbr. IS) does not use partitioning of the sets and bases the predictions on a single isotropic variogram model for the entire study area. Finally, the fourth approach (abbr. AS), differs from IS in that anisotropy parameters are estimated (for the entire domain) and used to transform into an isotropic coordinate system before the omni-directional variogram is estimated and fitted.

4.2 Spatial Model Parameter Estimation

Isotropic variogram models are estimated from the empirical omni-directional variogram of the training set by means of weighted least squares (WLS) fits. For the approaches that include anisotropy, $(\hat{R}, \hat{\theta})$ are estimated for the training sets in G1 and G2. Bilinear interpolation, implemented by means of the `akima` package, is used to estimate GDR on the anisotropy estimation grids of each cluster. Examples of the interpolated field generated on the anisotropy estimation grids in each domain are shown in Fig. 6.5. In these plots, the GDR is measured on a logarithmic color scale; for reference note that the natural logarithm of the threshold is $\log(x_c = 250) = 5.5215$.

Since G1 contains a number of clusters (cf. Fig. 4.7), $(\hat{R}, \hat{\theta})$ are based on the cluster average of the gradient tensor as given by Eq. (4.3). For each realization, we test if the isotropic hypothesis is supported based on (4.4), and then an isotropy restoring coordinate transformation is performed. The range and sill of the omnidirectional variogram are estimated in the isotropic coordinate system using the R function `automap` [HPTH09]. For each training set, the optimal variogram is selected from among the exponential, Gaussian, spherical and Matérn models. The estimates $(\hat{R}, \hat{\theta})$ are then incorporated to obtain the anisotropic variogram model.

4.3 Spatial Interpolation and Cross-validation

The method of ordinary kriging (OK) is used for interpolation using the `gstat` package [Peb04]. Validation measures compare estimates with "true" values at the validation locations. The validation measures are obtained by

calculating (i) the spatial average over the validation set and (ii) an average over the sampling realizations. For example, the mean error (ME) is defined as:

$$ME = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{N_v}\sum_{i=1}^{N_v}\left[\hat{X}(\mathbf{s}_i^j) - X(\mathbf{s}_i^j)\right],$$

where $M = 50$, N_v is the number of sensor locations in the validation set, $\hat{X}(\mathbf{s}_i^j)$ denotes the OK prediction at $\mathbf{s}_i^j$, and the $\mathbf{s}_i^j$ represents the $i-$th sampling point in the $j-$th realization. Similarly to ME, we define the mean absolute error (MAE), mean absolute relative error (MARE), mean square root error (MSRE) and mean relative square root error (MRSRE). We also report the mean values of the linear correlation coefficient, r, of Spearman's rank correlation coefficient, ρ, and of Kendall's rank correlation coefficient, τ.

The cross-validation results are reported in Table 6.4. The following general tendencies can be observed. Partitioning of the training and validation points into "normal-value" and "extreme-value" domains improves performance compared to single-domain models. In addition, accounting for anisotropy leads to improved performance over isotropic modeling assumptions. For the least populous training set ($p = 90$), the two-domain, anisotropic model performs better than the other models, with respect to all validation measures. The two-domain, isotropic model follows in performance.

The classification of performance is less definite for the larger training sets, since the isotropic models can achieve better values of certain measures (MAE, RMSE, RMSRE and r) than the anisotropic ones. This tendency is more prominent for the single-domain than for the two-domain models. In all cases the anisotropic models have a lower bias (in absolute value) and a lower MARE than their isotropic counterparts. The hierarchical correlation coefficients ρ and τ are consistently higher for the anisotropic models, in contrast with the linear correlation coefficient. The non-parametric coefficients are more reliable measures of correlation given the large deviation of the data set from Gaussian behavior.

Overall, the domain partitioning improves the validation measures. The only exception is the two-domain partitioning with isotropic variogram for $p = 60$, which leads to very high errors. Similar behavior is observed for the single-domain models at $p = 90$. These errors are due to points near the boundary of the plume, the values of which are under- (over-) estimated. These large errors appear in a few, out of the 50 configurations generated. While we cannot claim with absolute certainty that incorporating anisotropy

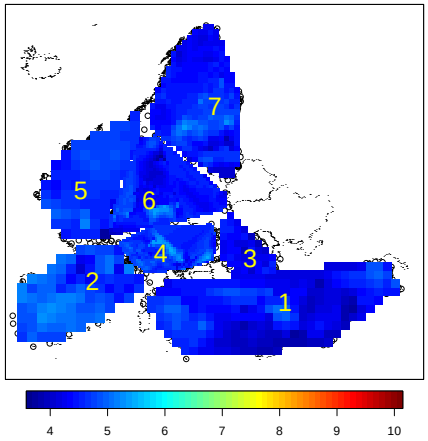
eliminates such boundary problems, the evidence (based on the 200 [total] training configurations investigated here) is that the coupling of domain partitioning with anisotropy modeling avoids such instabilities.

4.4 Interpolated Maps

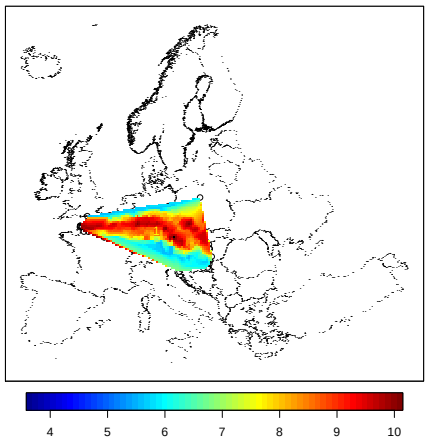
Interpolated values of the GDR distribution on a map grid containing 8000 nodes are generated using one (randomly chosen) of the $p = 90$ training configurations. The training sensor locations and the logarithms of GDR values at these locations are shown in Fig. 6.6. The visual analysis of the kriged maps that follows should be considered in connection with the quantitative cross-validation results listed in Table 6.4.

The kriged maps are shown in the plot table of Figure 6.7. The top row shows maps created using two-domain partitioning, while the maps in the bottom row are based on a single domain. The maps in the left column use an isotropic variogram model assumption, while the maps in the right column use an anisotropic variogram model. Blank (white) patches correspond to areas where the OK predictions are negative, leading to non-numeric logarithms. Differences between the single- and two-domain approaches are easily discerned: the former tends to over-estimate the background and smoothes excessively the south-eastern tail of the plume. On the other hand, given the large range of the GDR values, differences are not easily distinguished between the isotropic and anisotropic models (left vs. right columns) in Fig. 6.7. The most distinctive difference is that the isotropic model predicts higher values along the Eastern parts of the plume. Both models predict high values for the Eastern area, but there are directional differences. In particular, the anisotropic model gives a clearer bending of the plume in the South-East direction, which is in better agreement with the observed behavior, as shown in Fig. 4.3.

In Fig. 6.8 we plot the differences between the isotropic and anisotropic models' predictions, using domain segregation in both cases. As shown in Fig. 6.8(a), the differences between the two models in the background domain are small, roughly in the range of -10 to 10 nSv/h. On the other hand, the differences in the plume domain, shown in Fig. 6.8(b), range from < -3000 to > 5000 nSv/h. Fig. 6.8(b) also shows that the isotropic model predicts higher (lower) values than the anisotropic model along the boundaries (center) of the plume.



(a) Interpolated GDR on anisotropy estimation grid in domain G1.

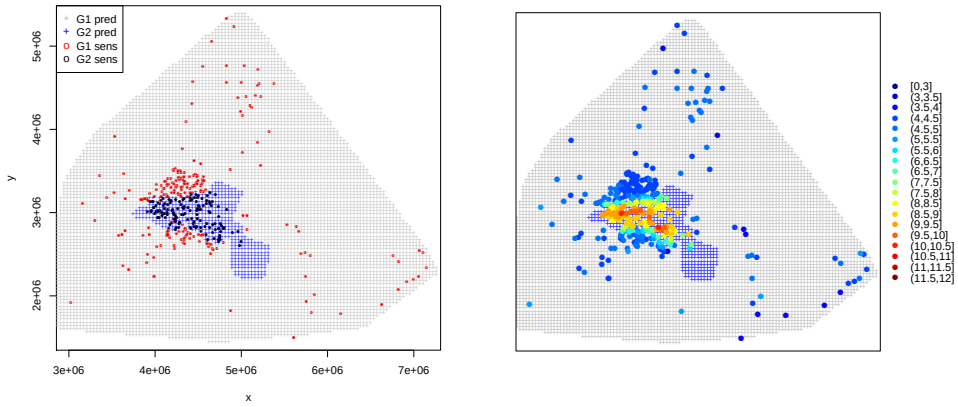


(b) Interpolated GDR on anisotropy estimation grid in domain G2.

Figure 6.5: Interpolated log(GDR) fields used in the clustered CHI anisotropy estimation.

		ME	MAE	MARE	RMSE	RMSRE	r	ρ	τ
$p = 10$	ID	-2.955	425.676	0.488	1144.454	3.023	0.965	0.920	0.795
	AD	-5.455	415.177	0.477	1148.318	3.031	0.965	0.921	0.797
	IS	-9.051	616.107	1.023	1369.577	4.686	0.950	0.806	0.644
	AS	-1.307	631.924	0.981	1417.986	4.735	0.947	0.812	0.649
$p = 30$	ID	-12.507	478.701	0.623	1263.680	4.615	0.957	0.915	0.785
	AD	-10.352	451.775	0.597	1238.962	4.656	0.958	0.917	0.789
	IS	-24.884	608.778	1.199	1389.701	5.663	0.947	0.804	0.641
	AS	-18.911	647.628	1.154	1467.948	5.788	0.941	0.808	0.644
$p = 60$	ID	-4448.786	13070.149	67.467	176246.953	1451.137	0.926	0.888	0.754
	AD	-12.335	511.034	0.739	1355.581	5.730	0.951	0.903	0.768
	IS	-31.959	645.676	1.521	1463.860	6.529	0.942	0.781	0.615
	AS	-24.276	652.488	1.418	1488.173	6.646	0.940	0.790	0.624
$p = 90$	ID	36.618	1032.371	1.616	2763.816	10.767	0.832	0.824	0.673
	AD	3.900	1007.464	1.599	2621.611	9.532	0.852	0.832	0.681
	IS	3141.611	15769.644	113.482	67314.193	623.990	0.870	0.710	0.537
	AS	802.443	99761.167	811.333	516659.990	5140.070	0.851	0.694	0.529

Table 6.4: Cross-validation measures calculated over the validation points for different partitions $p = (10, 30, 60, 90)$ of the GDR data set and different spatial modeling approaches. ID: Two-domain partitioning with isotropic variograms. AD: Two-domain partitioning with anisotropic variograms. IS: Single domain with isotropic variogram. AS: Single domain with anisotropic variogram. Validation measures represent means over 50 realizations of spatially averaged statistics. Numbers are rounded to the third decimal place. ME: Mean error. MAE: Mean absolute error. MARE: Mean absolute relative error. MRSE: Mean root square error. MRSRE: Mean root square relative error. r: mean linear correlation coefficient. R: mean Spearman correlation coefficient. τ : mean Kendall's tau.



(a) Training locations (b) Training values

Figure 6.6: (a) Partitioning of the sensor training locations between the "normal-value" (G1, red dots) and "extreme-value" (G2, black dots) sets. (b) Natural logarithm of GDR values at the training locations. (a,b) The map grid is partitioned into "normal-value" (G1, gray crosses) and "extreme-value" (G2, blue crosses) domains.

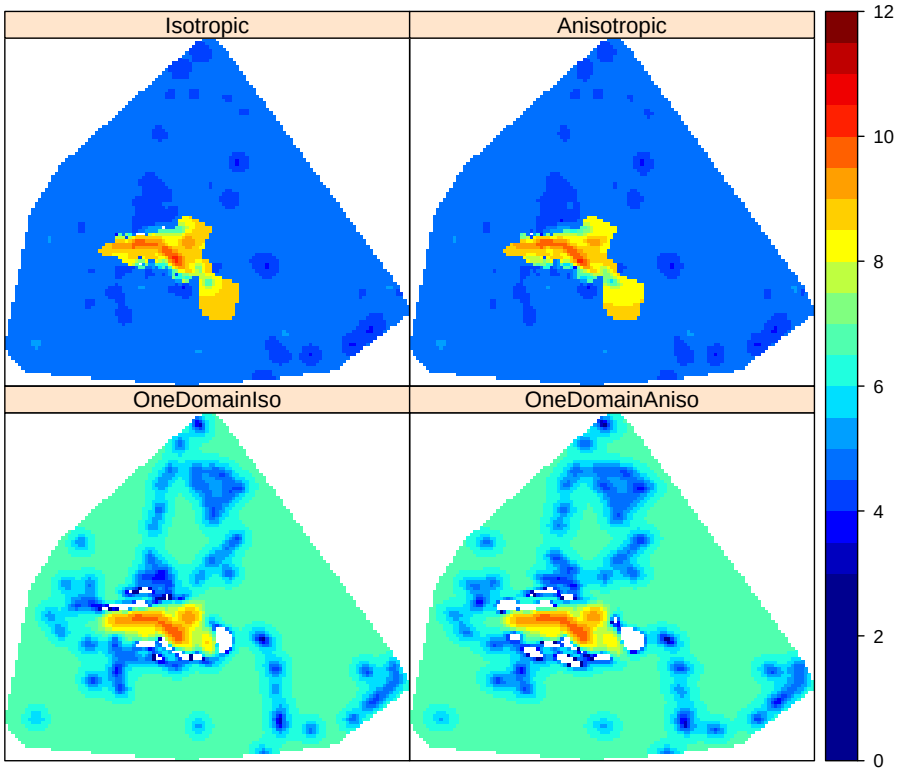
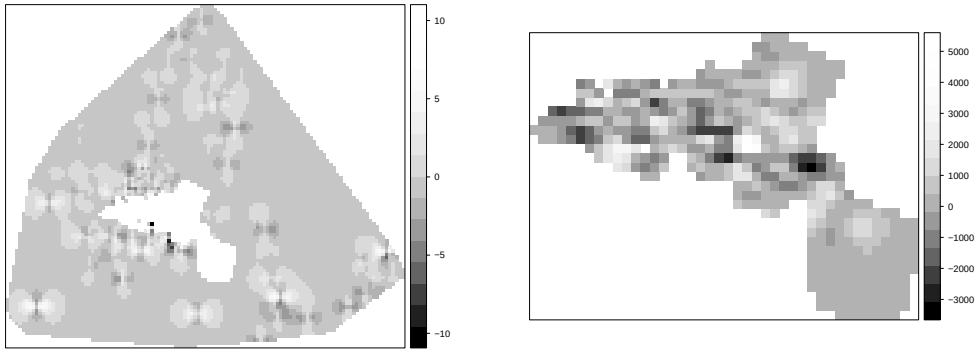


Figure 6.7: Interpolated GDR values with (top) and without (bottom) domain partitioning, using anisotropic (right) and isotropic(left) variograms.



(a) Difference of OK predictions in G1.

(b) Difference of OK predictions in G2.

Figure 6.8: Plots of the difference in Ordinary Kriging GDR predictions between the isotropic and anisotropic models. The interpolation is performed separately on the background (G1) and the plume (G2) domains.

Chapter 7

Application of CHI method on MRI data

In this chapter we investigate the application of *CHI* on Magnetic Resonance Imaging (MRI) data. Nowadays medical imaging applications increasingly attract the interest of the scientific community. MRI data have proved useful for imaging brain diseases. In the following chapter, we present some of the basic data types used in the literature and their connection to anisotropy estimation. We discuss the application of *CHI* to various data types. Finally we present application of *CHI* on MRI data, by means of a moving window procedure, to provide local estimates of water concentration anisotropy . Water concentration in the brain structure is highly connected to the diffusion of cancer cell's, therefor local estimates of anisotropy may provide useful information on tumor modeling.

Methods developed in Geostatistics aim to solve problems dealing with the characterization of spatial and spatio-temporal phenomena. Most of these phenomena emerge in scientific fields as mining, hydrology, meteorology, oceanography and environmental monitoring systems. On one hand, most observations are distributed over macro or earth-scale domains. On the other hand, the underlying scale is not a requirement for applying Geostatistics.

1 Magnetic Resonance Imaging

Magnetic Resonance Imaging, or MRI, is a method of imaging the interior of structures non-invasively. An MRI device consists of a magnet, magnetic gradient coils, an RF (radio frequency) transmitter and receiver, and a computer that controls the acquisition of signals and computes the MR images.

An atomic Nucleus is originally exposed to a static Magnetic field, absorbing energy. This energy is released in the form of a photon (resonates) when a varying electromagnetic field is applied at the proper frequency. An Image is computed from the resonance signals of which the frequency and phase (timing) contain space information. In typical MRI images the intensity is provided by the concentration, or density of observed nucleus and the exponential relaxation times of the signals following the transient electromagnetic field [DL08] (see also subsection 1.1). The early work of Damadian in [Dam71] showed that different tissues contain different amounts of water and exhibit different water proton relaxation times. Therefor MRI images may be considered as water concentration maps.

Water diffusion is strongly related to various disease processes. Structural barriers inside tissues, cause anisotropic water diffusion. Hence anisotropy is essential for understanding the abnormal development of diseases processes [Bea02]. Therefor, anisotropy estimation of water concentration may provide important advances in modeling water diffusion related diseases. Anisotropy estimation from MRI data has already captured the interest of the scientific community. Later on, in subsection 1.2 we shortly comment on some important studies related to anisotropy estimation.

1.1 Data acquisition

In typical MRI images, the observed intensity (weights) $S(TR, TE, G)$ for the specified repetition time (TR), exposure time (TE) and gradient strength of the magnetic field G for magnetic resonance data is expressed in [MKAN91] as

$$S(TR, TE, G) = S(\inf, 0, 0)e^{-TE/T2}(1 - 2e^{-(TR-TE/2)/T1}) + e^{-TR/T1}e^{-\gamma^2 G^2 \delta^2 (\Delta - \delta/3) ADC})$$

where S(inf,0,0) is the signal at TE=0, TR=inf. The terms TR,TE,T1 and T2 are the repetition time, the exposure time, the spin-lattice and spin-spin relaxation times, respectively ; γ is the gyromagnetic ratio, δ and G are the duration and strength of the gradient pulse, is the duration of the pulse and Δ is the time between on two pulses. Finally, the term ADC is the apparent diffusion coefficient which is determined from

$$\ln[S(TE, G)/S(TE, 0)] = -\gamma^2 G^2 \delta^2 \left(\Delta - \delta/3\right) ADC = -bADC$$

The direction of the diffusion sensitizing gradients can be controlled and the apparent diffusion coefficient (ADC) can be measured along the respective direction.

Based on the equations above and in some cases contrast agents it is possible to extract different attributes by controlling the TE,TR and the number of sensitizing gradients. Using positive contrast agents (Gadolinium), and $TR < 750ms$ and $TE < 40ms$ it is possible to acquire spin-lattice (T1) weighted images. Using negative contrast agents (SPIO), and $TR > 1500ms$ and $TE > 75ms$ it is possible to acquire spin-spin (T2) weighted images. Rearranging the second equation it is possible to calculate a single ADC value for each voxel in the brain. Using ADC values as color values it is possible to construct Diffusion Weighted Images (DWI) for a specified direction. Finally, applying various diffusion sensitizing gradients it is possible to sample enough data in order to define a second-order diffusion tensor (DT-MRI) that captures the diffusion properties of water along specific directions and represent it in the form of a tensor. [BAM03].

1.2 Related Work

In this section we examine how anisotropy may be incorporated in Tumor Modeling using different types of brain data. We are going to concentrate on MRI (T1 & T2), ADC and DT-MRI data.

Moseley et al in [MCK+90] examined the relationship between the diffusion behavior of water protons in normal gray and white matter of living cats with the diffusion-gradient strength and direction. In this particular work the anisotropy in each voxel is characterized by the ratio of the differences and sums of ADCs with diffusion-sensitizing gradients applied in two perpendicular directions; e.g x and y :

$$\frac{ADC_x - ADC_y}{ADC_x + ADC_y.} \tag{7.1}$$

Douek et al proposed in [DTP+91] a different approach. The anisotropy in each voxel was characterized by the ratio of two apparent diffusion constants (ADCs) measured with diffusion sensitizing gradient in two perpendicular directions:

$$\frac{ADC_x}{ADC_y}. \tag{7.2}$$

This ratio was later displayed as a color image. In [CCHR76] it was proposed that in white matter voxels, where this ratio appeared to be in its maximum value, corresponds to the ratio of the perpendicular and parallel fiber tract direction $ADC_\perp/ADC_{||}$.

Gelderen et al proposed in [vGdVD+94] a scalar anisotropy index that is proportional to the standard deviation of three $ADCs$ measured in three mutually perpendicular directions ADC_x, ADC_y and ADC_z divided by their mean value $\langle ADC \rangle$:

$$\frac{\sqrt{(ADC_x - \langle ADC \rangle)^2 + (ADC_y - \langle ADC \rangle)^2 + (ADC_z - \langle ADC \rangle)^2)}}{\langle ADC \rangle}. \quad (7.3)$$

Basser and Pierpaoli in [BP96] proposed a new set of quantitative parameters derived form the effective Diffusion Tensor D.

$$\mathbf{D} = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{bmatrix}.$$

First, they address a decomposition of the original tensor D into isotropic and anisotropic tensors:

$$\mathbf{D} = \underbrace{\langle \mathbf{D} \rangle \mathbf{I}}_{isotropic} + \underbrace{\mathbf{D} - \langle \mathbf{D} \rangle \mathbf{I}}_{anisotropic}$$

where $\langle \mathbf{D} \rangle = \frac{Trace(\mathbf{D})}{3}$ and $\mathbf{I}$ is the isotropic identity tensor. The anisotropic tensor is called deviatoric or diffusion deviation tensor $\mathbf{D'}$ as it measures the deviation of $\mathbf{D}$ from the isotropic tensor $\langle \mathbf{D} \rangle$:

$$\mathbf{D'} = \mathbf{D} - \langle \mathbf{D} \rangle \mathbf{I}$$

For the isotropic tensor the magnitude[1] of tensor $\langle \mathbf{D} \rangle \mathbf{I}$ is:

$$\sqrt{\langle \mathbf{D} \rangle \mathbf{I} : \langle \mathbf{D} \rangle \mathbf{I}} = \langle \mathbf{D} \rangle \sqrt{\mathbf{I} : \mathbf{I}} = \langle \mathbf{D} \rangle \sqrt{3}$$

and for an anisotropic tensor $\mathbf{D'}$ it is shown that

$$\sqrt{\mathbf{D'} : \mathbf{D'}} = \sqrt{(\lambda_1 - \langle \mathbf{D} \rangle)^2 + (\lambda_2 - \langle \mathbf{D} \rangle)^2 + (\lambda_3 - \langle \mathbf{D} \rangle)^2} = \sqrt{3Var(\lambda)}$$

[1] In order to measure the magnitude of the tensor $\mathbf{T}$ it is used the square root of the generalized tensor product or tensor dot product $\sqrt{\mathbf{T} : \mathbf{T}}$:

$$\mathbf{T} : \mathbf{T} = \sum_{i=1}^{3} \sum_{j=1}^{3} \mathbf{T_{ij}} = \sum_{i=1}^{3} \lambda_i^2$$

where λ denotes the eigenvalue of the T.

The Relative Anisotropy (RA) and Fractional Anisotropy (FA) are proposed in the same work as the following and dimensionless measures of anisotropy.

$$RA = \frac{\sqrt{\mathbf{D}' : \mathbf{D}'}}{\sqrt{\langle \mathbf{D} \rangle I : \langle \mathbf{D} \rangle I}} = \frac{\sqrt{\mathbf{D}' : \mathbf{D}'}}{\sqrt{3}\langle \mathbf{D} \rangle} = \frac{\sqrt{Var(\lambda)}}{E[\lambda]}$$

$$FA = \sqrt{\frac{3}{2}} \frac{\sqrt{\mathbf{D}' : \mathbf{D}'}}{\sqrt{\mathbf{D} : \mathbf{D}}}$$

2 Application of *CHI* as applied to MRI data

Tumor Modeling from MRI data is a relatively new scientific field but it is becoming increasingly prominent. Important advances is this field have followed Diffusion Tensor Imaging (DTI) development. However, DTI labs and data are very rare due to extremely high costs of DTI equipment. One important advancement from conventional MRIs is that a Diffusion Tensor includes anisotropy information as shown in [BP96]. Therefor it is interesting to discuss if the CHI method can be applied to MRI data. However, before applying the *CHI* method to magnetic resonance data there are several issues to be discussed.

Random process As explained in the previous sections, *CHI* is designed to estimate anisotropy parameters of random fields. We have explained that *CHI* provides a single value for anisotropy for the entire domain of interest. This means that we have to accept that the way cancer cells diffuse throughout the brain is a random process, if the CHI method is to work. This constraint is not completely satisfied as there are structural barriers in the brain whose sizes are not necessarily small compared to the domain size. On the one hand, we may accept that within a small area of the brain the cancer cells' movement is a random process, and that structural barriers consist part of this random process. On the other hand, as the area of interest decreases, the number of observation data also decreases given a fixed resolution of the MRI machine. The *CHI* method consists of a statistical analysis of the data. Therefore, it is important to keep in mind, though, that as the input data size decreases, several accuracy problems occur that may lead to either isotropic or extremely anisotropic estimates.

Microscopic scale Statistical analysis at a microscopic scale usually is bounded from the MRI resolution ability. We refer to a microscopic scale when the outer bounds of the area of interest consists the area of an MRI-voxel. In order to estimate inter-voxel anisotropy, it is essential to acquire

inter-voxel statistics related to the inter-voxel field' gradients. Unfortunately magnetic resonance imaging (MRI) data provide only a single value per voxel. This value represents the average value of the metered property per voxel. This restriction excludes MRI data from microscopic anisotropy analysis by estimating anisotropy directly from the data. However, these datasets could provide estimates for anisotropy per voxel by means of a moving window procedure (see section 4). This approach presumes that the statistics of the neighborhood, centered over an arbitrary pixel, fully represents the statistics of the specified voxel. Gradient data acquisitions such as DWI and DT-MRI have been used in the past in order to model anisotropy.

Diffusion Tensors vs Slope Tensors At the core of the *CHI* method is the slope tensor. The expected slope tensor $\hat{Q}_{ij}$ for Gaussian Random Fields (GRF) is linked to the Covariance Hessian Matrix H_{ij} thought the Covariance Hessian Identity [Swe62]. Slope tensors represent fully the second order derivatives of the covariance function of a GRF and provide a global anisotropy estimate of a given field. On the other hand, Diffusion Tensors are local estimates per voxel. The MRI Diffusion Tensors represent the magnetic diffusion gradient based on the attenuation of voxels through repeated scans.

The relation between the MRI Diffusion Tensor and the Slope Tensor is unclear to date. This relation merits further study as the DT-MRI is a complicated procedure and in order to define the relation, knowledge of each stage of the procedure is needed.

3 Dimensions The DT-MRI data provide an 3 by 3 diffusion tensor $\mathbf{D}$ for each voxel. Under the assumption that the diffusion tensor $\mathbf{D}$ can fully represent the second order derivatives of the covariance function; i.e $\mathbf{\hat{Q}_{ij} = D_{ij}}$, we could apply the *CHI* method to estimate meaningful anisotropy parameters per voxel. However, since there is no analytical solution for anisotropy parameters in three-dimensions, so one should approach this subject by minimizing the equation 3.8.

$$D = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{bmatrix} \xrightarrow{eq:3.8} [R, \theta] = (R_{21}, R_{31}, \theta, \phi, \psi),$$

given that θ, ϕ & ψ are the Euler angles for a three-dimensional Euler rotation matrix (see Appendix A) and R_{21}, R_{31} are ratios of the principal correlation axis.

Even though the theoretical background exists, this optimization problem suffers from multiple local minimums and has not been adequately solved yet.

Some pre-limitary research on three-dimensional analysis can be found in the Appendix A.

2 Dimensions In various medical imaging application, among them and MRI, the three dimensional data are acquired from vertically aligned slices. Even though the final result provides a three dimensional representation of the measured quantities, the data acquisition is performed in two dimensions. Therefor, it is not always valid to assume that the correlation of samples in the direction of the third axis is captured in data. In this case we should perform anisotropy analysis in two dimensions instead of three. In case we focus on 2D images (e.g slices) the *CHI* method can be applied under the assumption ($\hat{\mathbf{Q}}_{\mathbf{ij}} = \mathbf{D}_{\mathbf{ij}}$):

$$D = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix} \xrightarrow{eq:A.2,A.3} [R, \theta]$$

2.1 Relation between anisotropy parameters and diffusion in 2D tensor

In 2D the relation between anisotropy parameter and slope tensors are given from [CH08].

$$D_{11} = \frac{\sigma_x^2 \zeta^2}{\xi_1^2}(\cos^2\theta + R_{(2)1}^2 \sin^2\theta)$$

$$D_{22} = \frac{\sigma_x^2 \zeta^2}{\xi_1^2}(\sin^2\theta + R_{(2)1}^2 \cos^2\theta)$$

$$D_{12} = D_{21} = \frac{\sigma_x^2 \zeta^2}{\xi_1^2}(\sin\theta\cos\theta + (1 - R_{(2)1}^2)$$

where $\zeta = (1/2)\Delta\widetilde{c_x}(0)$, $\widetilde{c_x}(0)$ is the isotropic covariance function evaluated at zero lag, ξ_1 is the correlation length of the direction originally selected as major and σ_x^2 is the variance.

2.2 Relation between slope and diffusion tensor in three dimensional space with aligned vertical axis.

In the special case where in a 3-dimensional system the one of the correlation axis is aligned with the vertical axis of the coordinate system we may write explicitly the anisotropy parameters. It is safe assume anisotropic behavior

between slices equal to the anisotropy of the major or minor axes (see also appendix A.1.1). The remaining diagonal element should be either:

$$D_{33} = \max(\mathrm{D}_{11}, \mathrm{D}_{22}) \quad \text{or} \quad \min(\mathrm{D}_{11}, \mathrm{D}_{22})$$

assuming that $D_{13} = D_{31} = D_{23} = D_{32} = 0$ which gives a tensor of the form :

$$D = \begin{bmatrix} D_{11} & D_{12} & 0 \\ D_{21} & D_{22} & 0 \\ 0 & 0 & D_{33} \end{bmatrix}$$

Again for $D_{ij}, i, j \in [1, 2]$ the equations are the same as in 2D case and the addition term D_{33} can be easily estimated (see also Appendix A):

$$d_{\mathtt{diag}} = \frac{D22}{D11}, \quad q_{\mathtt{off}} = \frac{D12}{D11}$$

$$\theta = \frac{1}{2}\tan^{-1}\left(\frac{2q_{\mathtt{off}}}{1 - q_{\mathtt{diag}}}\right) \tag{7.4}$$

$$R_{2(1)} = \sqrt{1 + \frac{1 - q_{\mathtt{diag}}}{q_{\mathtt{diag}} + (1 + q_{\mathtt{diag}})\cos^2\theta}} \tag{7.5}$$

and

$$R_{(3)1} = \sqrt{\frac{D33}{D11}} \tag{7.6}$$

3 Spatial Interpolation Comparison Dataset (SIC2004) and MRI data

The variable used in the SIC 2004 exercise is natural ambient (background) radioactivity measured in Germany. The data, provided kindly by the German Federal Office for Radiation Protection (BfS), are gamma dose rates reported by means of the national automatic monitoring network (IMIS). In the frame of SIC2004, a rectangular area was used to select 1008 monitoring stations (from a total of around 2000 stations). The exercise consists of using 200 measurements (background) to estimate the values observed at the remaining 808 locations. In addition, a emergency data set was released, which contains an anomaly. The anomaly was generated by a simulation model, and does not represent measured levels [SIC].

We demonstrate anisotropy estimation on those datasets. Anisotropy changes at the emergency scenario even if the majority of the observed data remains the same. We may notice (figure 7.1) that the background scenario demonstrates an isotropic behavior that can be seen both from the *CHI* estimates and the semivariogram. On the other hand the anomaly that was introduced into the data set changes both the direction and the ratio of anisotropy. This behavior is also seen in the semi-variogram (figure 7.1). The best approach to this issue would be to deal with the anomaly separately, as it is not part of the background spatial process.

	Emergency	Background
$\hat{R}$	1.4564	1.1469
$\hat{\theta}$	49.9005	-39.7459

Table 7.1: Emergency and Background scenario *CHI* anisotropy estimates

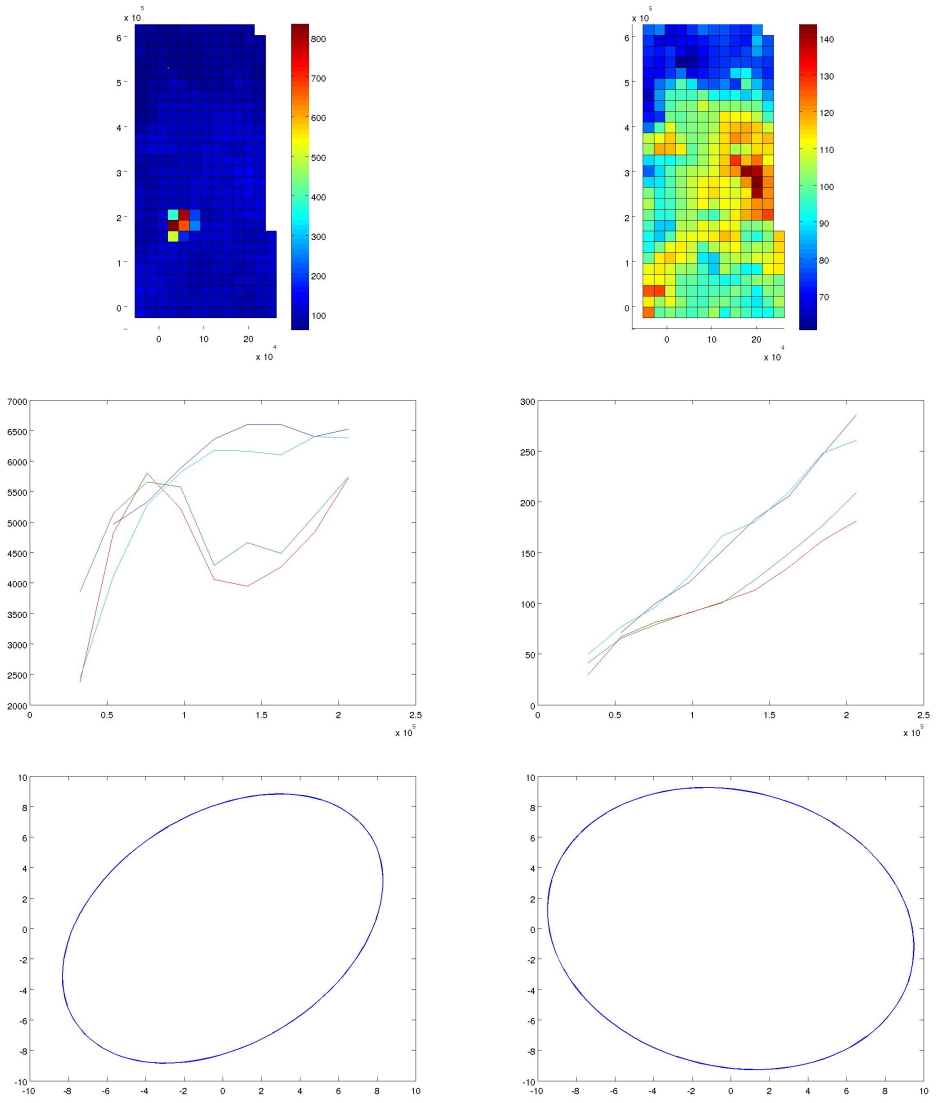


Figure 7.1: Sic2004 dataset. Top: linear Interpolation result for emergency (left) and background (right) scenarios. Note here different value range for the two scenarios (see colorbars) . Middle: The semi-variograms on 22.5(blue), 67.5(green), 112.5(red), 157.5(cyan) degrees from x axes of the emergency and background scenario respectively. Bottom: Ellipses semi-axes representing the ratio and direction of anisotropy *CHI* estimates. The axes lengths do not correspond to the true correlation lengths

In comparison to spatial data of SIC2004 scenario, we demonstrate the anisotropy estimates from an MRI images derived by *CHI* application (figures 7.2, 7.3, 7.4).The brain data shown in these figures, are selected MRI slices (4,5,8) from a tumor-modeling scenario (240x240x14). The analysis is performed only around the tumor area; due to morphological symmetry of the brain we selected to present only the right part of the tumor area in the following figures; equivalent results can be produced for any window area of the brain.

Even though, there is no obvious relation of the MRI images with the monitoring data presented earlier, the MRI images provide spatially correlated data in micro scale. The MRI images consist maps of the water concentration inside the brain. The water movement and the spatial dependence of the water concentration samples (that are provided from MRI images) are defined from the brain anatomy. Given that, MRI images provide spatially distributed samples, it is reasonable to examine whether the correlation of these samples changes differently in certain directions.

The MRI intensity images presented on the top-left of the figures are used to map white and gray matter in the brain. It is known that cancer cells diffuse differently in white and gray matter due to different water concentration. The concentration images shown on the top-right, present the simulated concentration of low-glioma cells (brain cancer cells) after 100 days of isotropic tumor diffusion. The extraction of white and gray matter is performed with threshold application on the intensity image. Every voxel that admits intensity value above 60 is marked as white matter; the remaining voxels are marked as gray matter.

We applied the *CHI* method on both the intensity and concentration images in order to compare the results (see figures 7.2, 7.3, 7.4 and table 7.2). It is clear that there is anisotropy in the area of interest in the intensity images (left column on figures). However the anisotropy is not present in the concentration image (right column on figures). This is verified both from the *CHI* estimates and the empirical semi-variograms in all figures. We believe that this difference in the behavior of the field is the result of the thresholding procedure. In addition in figure 7.4 we notice that the intensity image yields isotropy for the most of the area of interest. The results show significant departure from anisotropy. This anisotropic behavior exists mostly because of the dark spot on the top left of the intensity image that represents a solid structure inside this brain area. This area has completely different values from the (white) background and could be considered as an outlier. Consider now the effect on anisotropy estimates of the analogous emergency scenario. As shown by the emergency scenario above, outliers are able to greatly affect anisotropy and for this reason should be dealt separately as

in [SHPC]. However, there in not enough information to perform statistical analysis on such a small area.

slice	R_c	θ_c	R_i	θ_i
4	1.1671	5.6112	1.4958	79.6549
5	1.1427	-0.0517	1.7063	59.9122
8	1.209	-6.6583	1.4477	-38.7072

Table 7.2: *CHI* anisotropy estimates for slices 4,5 and 8 (shown later in figures 7.2, 7.3 and 7.4 respectively). The indices "c" and "i" denotes the concentration and intensity images respectively.

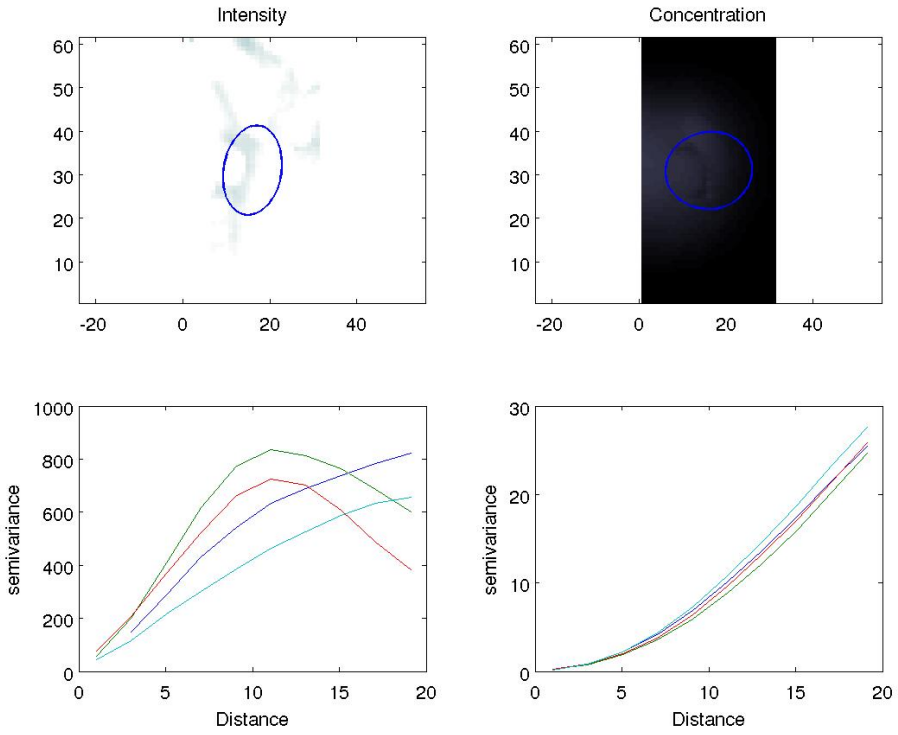


Figure 7.2: Slice 4 of the right part of the tumor area. Top-Right: Simulated concentration of tumor cells after 100 days and the ellipse that respects the *CHI* estimates for this simulation. Top-Left: Intensity MRI of the same area that was the initial model for the simulation. The ellipse again respects the *CHI* estimates for the intensity image. Bottom left and right: Directional semi-variogram for intensity and concentration respectively. The direction on the semi-variograms is 22.5(blue), 67.5(green), 112.5(red), 157.5(cyan) degrees from x axes.

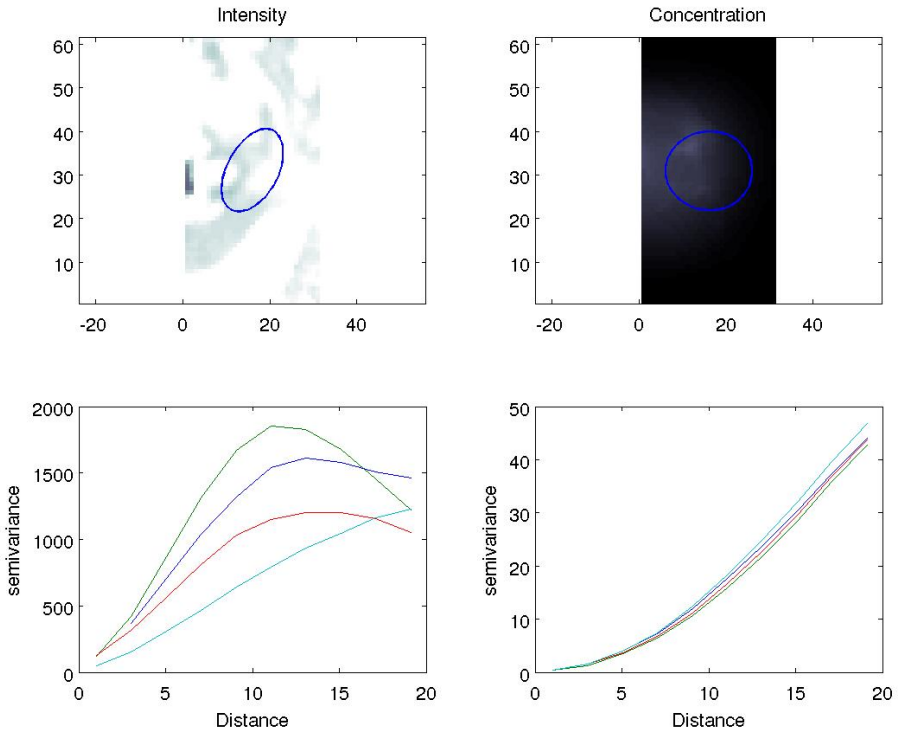


Figure 7.3: Slice 5 of the right part of the tumor area. Top-Right: Simulated concentration of tumor cells after 100 days and the ellipse that respects the *CHI* estimates for this simulation. Top-Left: Intensity MRI of the same area that was the initial model for the simulation. The ellipse again respects the *CHI* estimates for the intensity image. Bottom left and right: Directional semi-variogram for intensity and concentration respectively. The direction on the semi-variograms is 22.5(blue), 67.5(green), 112.5(red), 157.5(cyan) degrees from x axes.

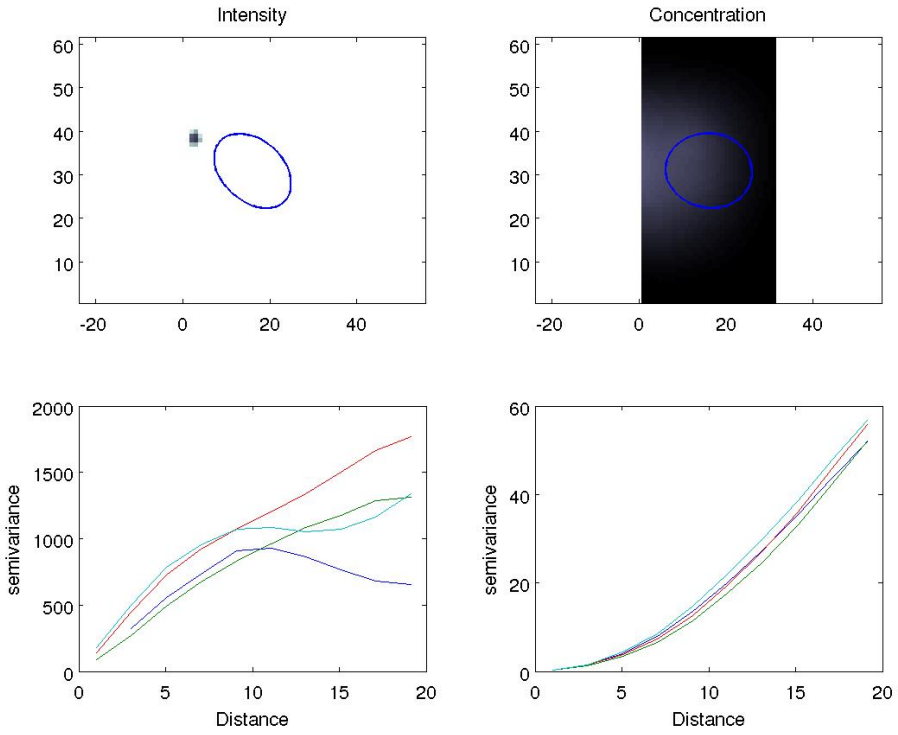


Figure 7.4: Slice 8 of the right part of the tumor area. Top-Right: Simulated concentration of tumor cells after 100 days and the ellipse that respects the *CHI* estimates for this simulation. Top-Left: Intensity MRI of the same area that was the initial model for the simulation. The ellipse again respects the *CHI* estimates for the intensity image. Bottom left and right: Directional semi-variogram for intensity and concentration respectively. The direction on the semi-variograms is 22.5(blue), 67.5(green), 112.5(red), 157.5(cyan) degrees from x axes.

4 Local *CHI* using moving Windows

So far we have presented the *CHI* method for global estimation of anisotropy in the area of interest. In case the area of interest is too small (e.g. a small part of the brain), then a global estimate of this area can be considered as a local estimate when it is examined from distance.

In the field of magnetic resonance imaging there is a rising interest in methods that can provide local estimates. To satisfy this need, we propose a local extension of the *CHI* method in terms of the moving window filter. This local method can be applied to MRI data. The local estimates of a cell are the *CHI* estimates of its vicinity, which is defined by the moving window size. The window moves across the area of interest and provides a set of overlapping neighborhoods, each neighborhood can be used then estimate anisotropy parameters that are assigned later on the center of the neighborhood (see figure 7.5). In any case, one should investigate whether the requirements of the *CHI* method are met for the window area (i.e. ergodicity and existence of a field's derivatives) and whether the window's size is large enough to capture changes of the field inside the brain. Under the assumption that the requirements are met for the window area, we provide local estimates. In the following sections we present application of local *CHI* using both synthetic and MRI data..

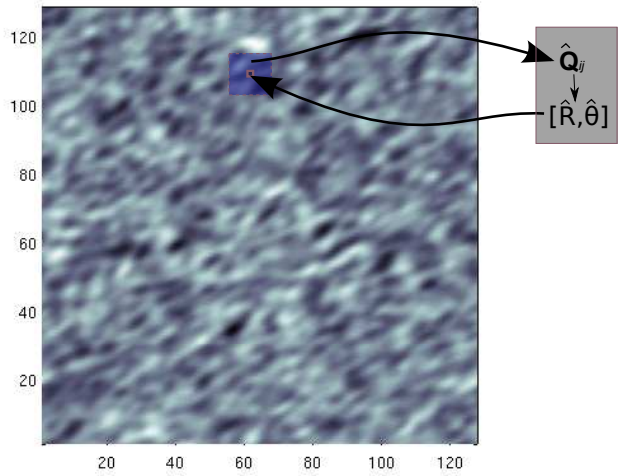


Figure 7.5: The local CHI anisotropy estimates are estimated using moving windows. The window moves across the area of interest and defines overlapping neighborhoods. Anisotropy parameters are estimated for each neighborhood and are assigned to its center.

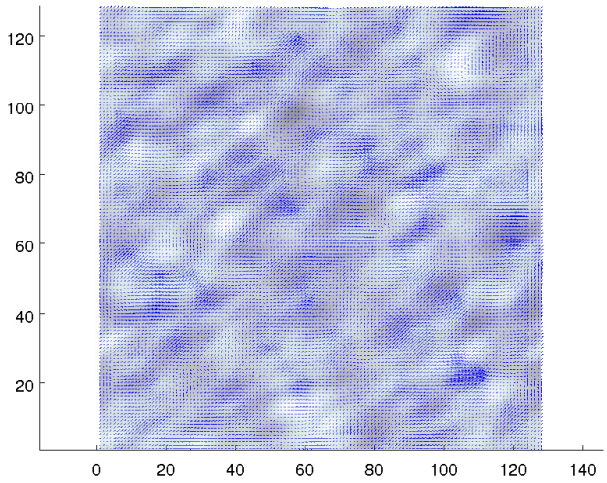


Figure 7.6: Left: GRF on a 128x128 lattice with anisotropy parameters $(R, \theta) = (2, 30°)$. The arrow's direction and relative length are equivalent to the moving window *CHI* etsimates. The window size is 10 and the major correlation length is 8.

4.1 Synthetic Data

We apply the moving window *CHI* on synthetic GRF N(0,1) with anisotropy parameters $(R, \theta) = (2, 30°)$. We have selected a square 10 by 10 window for the window filter. The length for the major and minor correlation axes is 8 and 4 respectively. The angle between the major correlation axes and the x-axis of the coordinate system is 30°. The generation of the GRF is performed using the FFT method described in section 5.1.

In figure 7.6 we see the realization of the GRF and an enlarged part of the same image in figure 7.7. We may notice on the enlarged part (figure 7.7) that the direction of the majority of the arrows agrees with the theoretical value of the synthetic GRF (30°). However there are parts that the direction of the arrows change. This behavior is justified based on the local fluctuation of the random field. This statement is also confirmed by the histograms of the anisotropy estimates (R, θ) (figure 7.8).

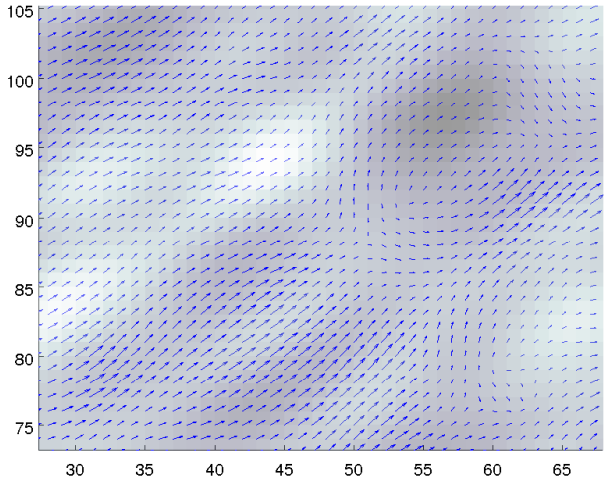


Figure 7.7: Enlarged image that demonstrates local anisotropy estimates (right) using arrows. The arrow's direction and relative length are equivalent to the moving window *CHI* etsimates. The window size is 10 and the major correlation length is 8.

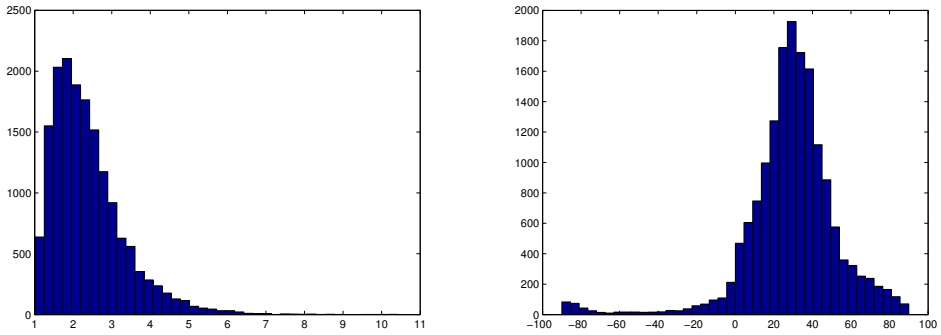


Figure 7.8: Histograms of the estimated local ratio $\hat{R}$ (top) and local direction $\hat{\theta}^{\circ}$ (bottom) for the synthetic data presented in figures 7.6 and 7.7. The GRF was generated with anisotropy parameters $(R, \theta) = (2, 30)$.

4.2 MRI data

We apply the local *CHI* on the 6^{th} slice of an MRI intensity image (figure 7.9). The MRI anisotropy estimates (blue arrows) appear to demonstrate the direction of the most correlated neighbor (details in the enlarged image 7.10). The length of the arrows represents the estimated anisotropy ratio. Based on the fact that the intensity represents the water concentration inside each pixel the following quiver plots demonstrate the local anisotropy estimates for water concentration inside the brain. This information is useful for brain-tractography and modeling various brain diseases that affect the water concentration in brain.

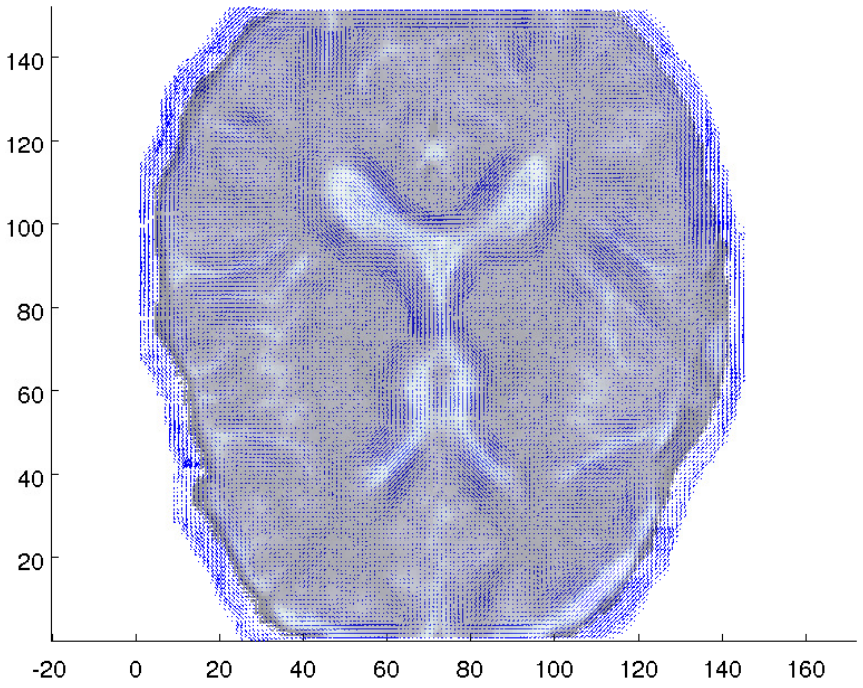


Figure 7.9: Local *CHI* local estimates of anisotropy for slice 6 of the MRI brain data. The arrow at each location demonstrates the direction of the principal correlation axis and the length of the arrow is analogous to the anisotropy ratio.

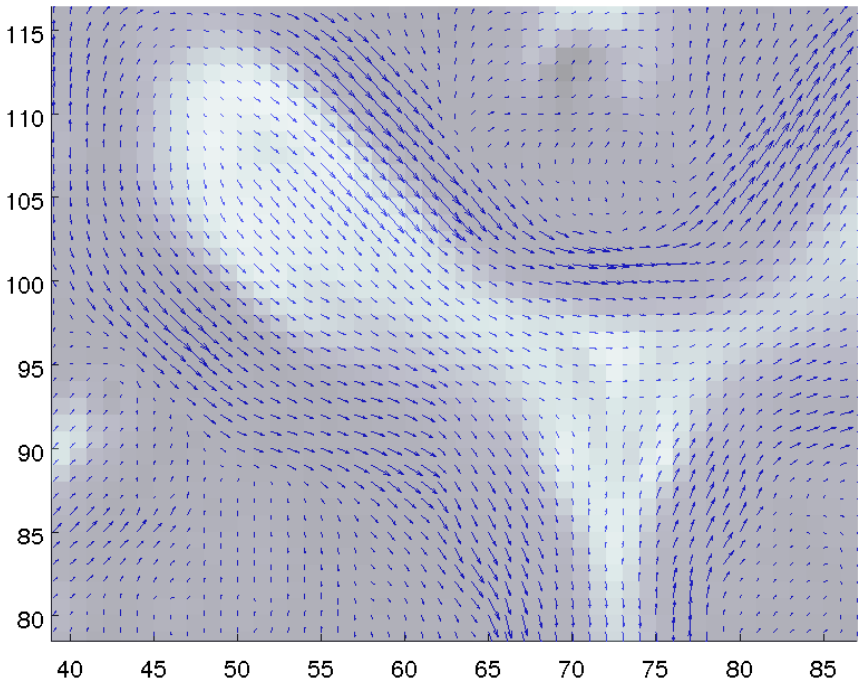


Figure 7.10: Enlarged part of figure 7.9. The arrow at each location demonstrates the direction of the principal correlation axis and the length of the arrow is analogous to the anisotropy ratio.

Chapter 8

Conclusions

From the perspective of environmental surveillance, there is a need for computationally efficient methods that can provide near real-time warnings for developing environmental threats. This thesis introduces the clustered CHI (*CCHI*) method for the estimation of geometric anisotropy parameters from scattered two-dimensional spatial data. The proposed method incorporates the computational efficiency of single-cluster CHI with a segmentation procedure, which can partition a heterogeneously sampled study area into smaller subsets to facilitate the spatial analysis. Based on our experience, the interpolation performance of clustered CHI improves when the sampling density, the "degree of stationarity" and the differentiability of the sampled process are increased.

To validate the above statement we presented some synthetic and real-case scenarios to compare the behavior of original *CHI* and *CCHI* algorithms. In addition, some other unsupervised clustering algorithms were also examined. An advantage of the *CCHI* clustering algorithm is that it does not require prior knowledge of method specific parameters. The *CCHI* algorithm was shown to behave well for several monitoring network scenarios without additional parameter tuning.

We illustrate the *CCHI* method by application to a "difficult" GDR data set which involves deviations from Gaussianity due to several factors, and significant variations of the sampling density across the study area. Application of *CCHI* leads to improved interpolation validation measures when compared to estimates that are based on the isotropic variogram hypothesis. The CHI method is computationally fast; for example, it requires only 0.17 seconds to estimate anisotropy in a domain containing around 2500 points, if the domain is treated as a single cluster. Segregation of the domain into clusters increases the computation time to about 9 seconds.

In addition to spatial statistics, other scientific fields that require estima-

tion of anisotropy could benefit from the application of *CHI* .

We investigated how *CHI* estimates can be used for medical imaging applications. We propose a local extension of *CHI* by means of a moving window fitler that gives meaningful results for local estimates of anisotropy. The distributions of anisotropy estimates for the moving window *CHI* are confirmed with visual inspection of synthetic GRFs. The application of moving window *CHI* on MRI data gives promising local estimates of anisotropy. Preliminary efforts to solve the *CHI* equations in three dimensions also provide a starting point for further discussion on *CHI* extensions. Three dimensional *CHI* can benefit both Geostatistics and medical imaging applications. However further validation against other types of Magnetic Resonance data has to be performed.

The R codes that we developed for the implementation of *CCHI* are part of the R packages `Intamap` and `IntamapInteractive`, which can be downloaded from
`http://sourceforge.net/projects/intamap/develop` or from the Intamap web site at: `http://www.intamap.org`.

Appendix A

Three Dimensional *CHI*

1 Appendix *CHI* method in 3D

From equation (17) in [CH08] we note

$$Q_{ij} = -\frac{R^2_{l(1)}}{\xi_1^2 d} \triangle c_x(0) U_{li}(\bar{\theta}) U_{lj}(\bar{\theta}) \quad \texttt{i,j=1...d} \tag{A.1}$$

where U_{ij} are the elements of the 3d rotation matrix (x'z'x') and $\bar{\theta}$ is an angle vector $\bar{\theta} = (\theta, \phi, \psi)$ as presented in wolfram.

$$U = BCD$$

Where:

$$D = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} B = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

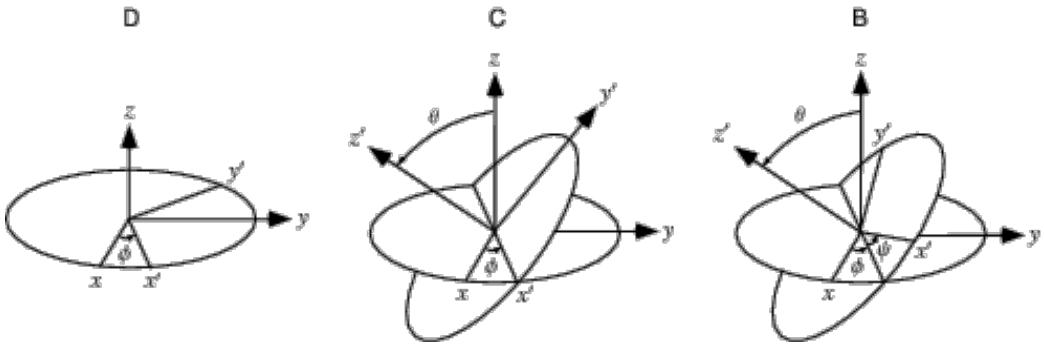


Figure A.1: Euler angles

So :

$$\begin{aligned}
u_{11} &= \cos\psi\sin\phi - \cos\theta\cos\phi\sin\psi \\
u_{12} &= \cos\psi\sin\phi + \cos\theta\cos\phi\sin\psi \\
u_{13} &= \sin\psi\sin\theta \\
u_{21} &= -\sin\psi\sin\phi - \cos\theta\cos\phi\cos\psi \\
u_{22} &= -\sin\psi\sin\phi + \cos\theta\cos\phi\cos\psi \\
u_{23} &= \cos\psi\sin\theta \\
u_{31} &= \sin\theta\sin\phi \\
u_{32} &= -\sin\theta\cos\theta \\
u_{33} &= \cos\theta
\end{aligned}$$

We may now rewrite equation (.A.1) as:

All together

$$\begin{aligned}
Q_{11} &= \tfrac{\sigma_x^2\zeta^2}{\xi_1^2}(u_{11}^2 + R_{2_{(1)}}^2 u_{21}^2 + R_{3_{(1)}}^2 u_{31}^2) \\
Q_{12} &= \tfrac{\sigma_x^2\zeta^2}{\xi_1^2}(u_{11}u_{12} + R_{2_{(1)}}^2 u_{21}u_{22} + R_{3_{(1)}}^2 u_{31}u_{32}) \\
Q_{13} &= \tfrac{\sigma_x^2\zeta^2}{\xi_1^2}(u_{11}u_{13} + R_{2_{(1)}}^2 u_{21}u_{23} + R_{3_{(1)}}^2 u_{31}u_{33}) \\
Q_{23} &= \tfrac{\sigma_x^2\zeta^2}{\xi_1^2}(u_{12}u_{13} + R_{2_{(1)}}^2 u_{22}u_{23} + R_{3_{(1)}}^2 u_{32}u_{33}) \\
Q_{33} &= \tfrac{\sigma_x^2\zeta^2}{\xi_1^2}(u_{13}^2 + R_{2_{(1)}}^2 u_{23}^2 + R_{3_{(1)}}^2 u_{33}^2) \\
Q_{22} &= \tfrac{\sigma_x^2\zeta^2}{\xi_1^2}(u_{12}^2 + R_{2_{(1)}}^2 u_{22}^2 + R_{3_{(1)}}^2 u_{32}^2)
\end{aligned}$$

We also have to set the ratio array to be $\bar{R} = (1, R_{2_{(1)}}^2, R_{3_{(1)}}^2)$

$$
\begin{aligned}
u_{11}^2 &= \cos^2\theta\cos^2\phi\sin^2\psi - 2\cos\theta\cos\phi\sin\psi\sin\phi\cos\psi + \sin^2\phi\cos^2\psi \\
u_{21}^2 &= \sin^2\psi\sin^2\phi + 2\sin\psi\sin\phi\cos\theta\cos\phi\cos\psi + \cos^2\theta\cos^2\phi\cos^2\psi \\
u_{31}^2 &= \sin^2\theta\sin^2\phi \\
u_{11}u_{12} &= \cos^2\psi\sin^2\phi - \cos^2\theta\cos^2\phi\sin^2\psi \\
u_{21}u_{22} &= \sin^2\psi\sin^2\phi - \cos^2\theta\cos^2\phi\cos^2\psi \\
u_{31}u_{32} &= -\sin^2\theta\cos\psi\sin\phi \\
u_{11}u_{13} &= \cos\psi\sin\phi\sin\psi\sin\theta - \cos\theta\cos\phi\sin^2\psi\sin\theta \\
u_{21}u_{23} &= -\sin\psi\sin\phi\cos\psi\sin\theta - \cos\theta\cos\phi\sin^2\psi\sin\theta \\
u_{31}u_{33} &= \sin\theta\sin\phi\cos\phi \\
u_{12}u_{13} &= \cos\psi\sin\phi\sin\psi\sin\theta + \cos\theta\cos\phi\sin^2\psi\sin\theta \\
u_{22}u_{23} &= -\sin\psi\sin\phi\cos\psi\sin\theta + \cos\theta\cos\phi\sin^2\psi\sin\theta \\
u_{32}u_{33} &= -\sin\theta\cos^2\theta \\
u_{13}^2 &= \sin^2\psi\sin^2\theta \\
u_{23}^2 &= \cos^2\psi\sin^2\theta \\
u_{33}^2 &= \cos^2\theta \\
u_{12}^2 &= \cos^2\theta\cos^2\phi\sin^2\psi + 2\cos\theta\cos\phi\sin\psi\sin\phi\cos\psi + \sin^2\phi\cos^2\psi \\
u_{22}^2 &= \sin^2\psi\sin^2\phi - 2\sin\psi\sin\phi\cos\theta\cos\phi\cos\psi + \cos^2\theta\cos^2\phi\cos^2\psi \\
u_{23}^2 &= \cos^2\psi\sin^2\theta
\end{aligned}
$$

If we want to write this problem in the using matrices then 1 can be written :

$$
\begin{bmatrix} Q_{11} \\ Q_{12} \\ Q_{13} \\ Q_{23} \\ Q_{33} \\ Q_{22} \end{bmatrix} = \frac{\sigma_x^2\zeta^2}{\xi_1^2} \begin{bmatrix} u_{11}^2 & u_{21}^2 & u_{31}^2 \\ u_{11}u_{12} & u_{21}u_{22} & u_{31}u_{32} \\ u_{11}u_{13} & u_{21}u_{23} & u_{31}u_{33} \\ u_{12}u_{13} & u_{22}u_{23} & u_{32}u_{33} \\ u_{13}^2 & u_{23}^2 & u_{33}^2 \\ u_{12}^2 & u_{22}^2 & u_{32}^2 \end{bmatrix} \begin{bmatrix} 1 \\ R_{2_{(1)}}^2 \\ R_{3_{(1)}}^2 \end{bmatrix}
$$

So in order to get the parameters estimates we need to divide each one of these equations with Q_{11} in order to reduce the number of unknown parameters. Someone may notice that the term $(\frac{\sigma_x^2\zeta^2}{\xi_1^2})$ is eliminated.
We may now use an optimization method in order to estimate the unknown parameter vector.

1.1 Special Case: One anisotropy principal axes aligned to the coordinate system

In the special case where the z-axes of the coordinate system is aligned with one of the anisotropy principal axes, then it is possible to provide closed form

solution for three-dimensional data. The rotation matrix U takes the form:

$$U = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

So equations in 1 take the form:

$$\begin{aligned} Q_{11} &= \tfrac{\sigma_x^2\zeta^2}{\xi_1^2}(cos^2\phi + R_{2_{(1)}}^2 sin^2\phi) \\ Q_{12} &= \tfrac{\sigma_x^2\zeta^2}{\xi_1^2}(\cos\phi\sin\phi - R_{2_{(1)}}^2 \sin\phi\cos\phi) \\ Q_{13} &= 0 \\ Q_{23} &= 0 \\ Q_{33} &= \tfrac{\sigma_x^2\zeta^2}{\xi_1^2} R_{3_{(1)}}^2 \\ Q_{22} &= \tfrac{\sigma_x^2\zeta^2}{\xi_1^2}(sin^2\phi + R_{2_{(1)}}^2 cos^2\phi) \end{aligned}$$

We may notice that everything besides Q33 is the same with 2D *CHI* . Normalizing everything with Q11 we can write:

$$q_{\texttt{diag}} = \frac{Q22}{Q11}, \quad q_{\texttt{off}} = \frac{Q12}{Q11}$$

$$\theta = \frac{1}{2}\tan^{-1}\left(\frac{2q_{\texttt{off}}}{1 - q_{\texttt{diag}}}\right) \tag{A.2}$$

$$R_{2(1)} = \sqrt{1 + \frac{1 - q_{\texttt{diag}}}{q_{\texttt{diag}} + (1 + q_{\texttt{diag}})\cos^2\theta}} \tag{A.3}$$

and

$$R_{(3)1} = \sqrt{\frac{Q33}{Q11}} \tag{A.4}$$

where $\theta \in [-\pi/2, \pi/2]$ and $\{R_2(1), R_3(1)\} \in [0, \infty)$.

Appendix B

Anisotropy Tutorial

Geostatistics Research Unit (GRU)

Technical University of Crete

Development of Geostatistical Software for Environmental and

Mining Applications

Estimation of Geometric Anisotropy by means of the Covariance Hessian Identity (CHI)

User guide for anisotropy estimation code in R

Topic	Anisotropy Estimation from Scattered Data in $2D$
Relevant Project	INTAMAP
Type of Project	STREP project funded by EC under FP6, PRIORITY IST-2005-2.5.12 ICT for Environmental Risk Management
Type of Document	Public
Authors	Giannis Spiliopoulos and Dionisis Hristopulos
Date	30 October 2010
Version	2.0

GRU director
Prof. Dionisis T. Hristopulos
Technical University of Crete, GREECE
E-mail: dionisi@mred.tuc.gr

http://www.mred.tuc.gr/home/hristopoulos/dionisi.htm

Abstract

This tutorial describes the basic information needed in order to perform anisotropy estimation using the `Intamap` and `IntamapInteractive` R packages. The main aspects covered in this tutorial are the package installation, a short description of the available methods for anisotropy estimation, and a number of examples for each method.

1 R basics

In this section we present some basic commands that you might find useful in case you are not familiar with the R language. A good starting point for learning R is the document "An introduction to R" that can be found under the *manuals* menu at http://www.r-project.org/. There is also a search engine for R at http://www.rseek.org.

1.1 Getting help

It it important to know how to use the help files for R functions. Placing a "?" in front of any command name prints on the screen the help file for that command. Examples for the commands "summary" and "plot" are shown below:

```
#help for the summary command
?summary

#help for the plot command
?plot
```

1.2 Data manipulation

R packages usually include sample data sets. In order to see datasets that have already been installed in your R system use the command

```
#print to screen available data sets.
data()

#print only base and sp datasets
data(package=c("base","sp"))
```

These datasets are saved in the form of a **data.frame** structure. In order to load a specific set in your workspace you should use the **data** command

```
#check available variables in your workspace
ls()

#load data.frame meuse
data(meuse)
```

```
#you can  now see the variable named meuse appearing in your workspace
ls()
```

We will describe how you can **insert your own data** as a data.frame in R. The following example demonstrates how to read data from a csv or txt file.

```
#read the help file of read.csv command if you think it is necessary
?read.csv

#reading data from a csv file and save it in a data.frame:
myData=read.csv(file="path/myfile.csv")

#Another command to read data from a csv (or a even a txt ) file:
myData=read.table(file="path/myfile.csv",sep=",",header=TRUE)

#An easy way to list the variables included in this R object is:
str(myData)

#The same operation is also possible with the following command:
summary(myData)
```

Note that is possible to insert data from a text file with a different ascii character separator, just by using a different separator in the "sep" argument in read.table command.

1.3 Using R packages

You can check which packages are installed in your computer by means of the command:

```
library()
```

The easiest way to extend the installed R functionality is to download and install packages from the CRAN server using the following command (e.g. to install package "sp"):

```
#install sp package from CRAN
install.packages("sp")

#To remove a package you should use the command:
remove.packages("sp")
```

After the package is downloaded to the computer, you have to load it in the workspace in order to use the functions that it includes. This is done as follows:

```
#load the already installed package sp.
library(sp)
```

2 Installation of the Intamap package

First make sure you have already installed the **latest R version** (current is 2.9). You can find the necessary information and software for installing R at http://www.r-project.org/.
After that you should download the **Intamap and IntamapInteractive** packages. At this moment these packages are not available from CRAN and the only available public source can be found at http://sourceforge.net/projects/intamap.
Click on **Develop** → **Code** → **SVN Browse** to get the source files. We suggest you download the full packages from the **pkg** folder. Alternatively, it is possible to download and modify the source code that can be found in the **intamap** and **intamapInteractive** folders.
Windows users could install the Intamap packages as a **local zip file**.

Packages → Install package(s) from local zip file()

For Unix users the command to install the source packages should be

R CMD INSTALL intamap.tar.gz

The most common problem during installation is that some necessary CRAN packages may not be installed on your computer. To solve this problem, you should start R and issue the following command at the command line:

```
install.packages(c('sp','gstat','akima','rgdal','automap','mvtnorm','evd'))
```

By this point you should have already installed the Intamap package. Using the following command inside the R command line, you should be able to load the intamap library.

```
library(intamap)                #loads to workspace intamap package
library(intamapInteractive)     #loads to workspace intamapInteractive package
?estimateAnisotropy             #print the help file
```

If you are familiar with R and do not wish to install the whole intamap package (recommended), you could extract the zipped files into a working directory and use only the files you need for anisotropy estimation (estimateAnistropy.R, doSegmentation.R, anisotropyChoice.R).

```
#set the working directory to the specified folder setwd('path')

# load at workspace the source files source('estimateAnisotropy.R')
source('doSegmentation.R') source('anisotropyChoice.R')
```

3 Interpretation of Anisotropy Estimation Results

The values of the parameters returned from the anisotropy estimation routines are defined in the help function of each command.Below we present some examples for illustration purposes.

The following script generates a synthetic gaussian field on a 128x128 grid using the *run.Gen* function, which is a custom function for generating synthetic fields, after we have first inserted the necessary parameters (R and Theta) through standard input (aka keyboard). Next we use estimateAnisotropyGrid intamap-internal function to estimate the parameters of the generated field and draw an ellipse according to the estimated parameters. Note that the length of the axis of the ellipse drawed here are note proportional to the actual correlation lenght of the field at each direction, and it is only useful in order to interpret the result.

```
#Set the values R and theta for the simulated field
R=as.numeric(readline("Set anisotropy Ratio :"))
theta=as.numeric(readline("Set anisotropy direction (degrees) :"))

#run.Gen generates a field ~N(0,1) on a grid 128x128
dat=run.Gen(theta=theta,xi1=R*4,xi2=4)

#estimateParameters
res=intamap:::estimateAnisotropyGrid(dat$x,dat$y,dat$z)

#plots an ellipse to illustrate the anisotropy parameters.
#As base we use an ellipse centered at (64,64) with the
#major axes  alligned to x axes and small axes  to be 20 grid cell
#length
plotEllipse(R=res$R,theta=res$theta,20,c(64,64))
print(res)
```

And the output looks like this :

```
#Anisotropic synthetic field
 #Anisotropy Ratio=3
 #Anisotropy
direction =-25 (deg)

#ANISOTROPY ESTIMATES
$R
[1] 2.9409

$theta.deg
[1] -24.331

$Q
     Q11      Q22      Q12
[1,] 493.51 1578.29 616.56

$doRotation
[1] TRUE
```

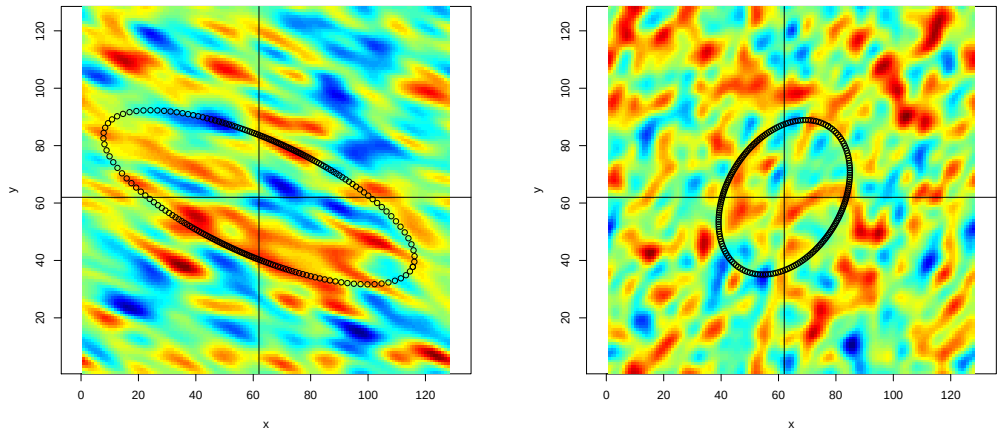
```
#Anisotropic synthetic field
#Anisotropy Ratio=1.5
#Anisotropy
direction= 60(deg)

#ANISOTROPY ESTIMATES
$R
[1] 1.45534

$theta.deg
[1] 58.34072

$Q
     Q11      Q22       Q12
[1,] 1531.45 1106.69 -422.61

$doRotation
[1] TRUE
```



(a) (b)

Figure B.1: (a) Synthetic Gaussian field with R =3 and theta =-25, (b) Synthetic Gaussian field with R =1.5 and theta =60

Assuming a Cartesian coordinate system of axes x and y, θ represents the angle between the horizontal axis and PA1, where PA1 is one of the principal axes of the ellipse, arbitrarily selected (PA2 will denote the other axis). R represents the ratio of the correlation along PA1 divided by the correlation length PA2. Note that the returned value of R is always greater than one (see 'value' below.)

ratio: The estimate of the anisotropy ratio parameter. Using the degeneracy of the anisotropy under simultaneous ratio inversion and axis rotation transformations, the returned value of the ratio is always greater than 1.

direction: The estimate of the anisotropy orientation angle. It returns the angle between the major anisotropy axis and the horizontal axis, and its value is in the interval (-90,90) degrees.

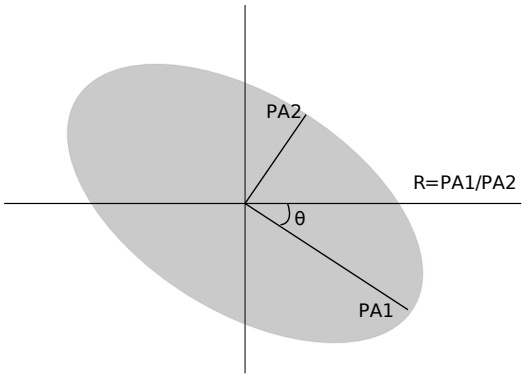


Figure B.2: Illustration of the anisotropy estimates.

4 Examples of Anisotropy Estimation

As stated above, there are two ways of using the anisotropy estimation routines. The first one is to install the intamap package and use the default values. The other option is to use selectively the internal anisotropy estimation functions of the package. In the second approach, one can define options such as different interpolation grids and interpolation methods (e.g., bicubic and biharmonics splines) for the estimation of sample derivatives.

Function	Package	Usage
Package functions		
estimateAnisotropy 4.1	INT	Provides single cluster anistotropy estimates for 2D data.
doSegmentation 4.1	INT-IN	Performs clustering for 2D data based on spatial and sampling density criteria.
anisotropyChoice 4.1,4.2.2	INT-IN	Wraps the two previous functions and provides anisotropy estimates for each cluster detected and a weighted average of the 2D dataset
Internal functions		
estimateAnisotropySc 4.2	INT	Single cluster for anisotropy estimates. This function is called by the *estimateAnisotropy* function.
segmentData 4.2.1	INT-IN	Segmentation-Clustering function. This function is called by the *doSegmentation* function.

Table B.1: Quick reference guide for the functions presented in the next sections. INT and INT-IN refer to the **Intamap** and **IntamapInteractive** packages.

4.1 Using the defaults

This is the easy approach for using the anisotropy code. It is based on the default (bilinear) interpolation method for all the cases. For single cluster anisotropy parameters estimation we have two different approaches. First approach if you only have a "SpatialPointsDataFrame" and the second one is to create an "Intamap" object. For the anisotropy estimation using clustering only the second approach is available.Here we use the Dataframes **meuse** and **quakes** which are included in the R distribution.

```
library(intamap)    #load intamap package
library(intamapInteractive) #load intamapInteractive package
data(meuse)     #load meuse data.frame

#here we change this data.frame to Spatial data frame
#Type  ``?coordinates'' to get more info.
coordinates(meuse)=~x+y

#To see what this data set involves type:
summary(meuse)

#For more info on the data set use:
?meuse

#To see a plot of the coordinates use
plot(meuse)

#INTAMAP PACKAGE
#-First approach
#single cluster for scattered data. Using a SpatialPointsDataFrame
params=estimateAnisotropy(meuse,"cadmium")
print(params)

#-Second approach
#single cluster for scattered data. Using an Intamap Object
object=createIntamapObject(observations=meuse,data=meuse$cadmium)
params=estimateAnisotropy(object)
print(params$anisPar)


#INTAMAP-INTERACTIVE PACKAGE
data(quakes)
```

```
coordinates(quakes)=~long+lat

#only Second approach is available:
object=createIntamapObject(observations=quakes,data=quakes$mag)
#multiple clusters: returns the cluster index, the anisotropy estimation for
#each cluster and the average (non-linear ) anisotropy estimates for the hole
#area.
params=anisotropyChoice(object)
print(params$anisPar)


#to plot the clustering result you may type the following commands.
plot(coordinates(quakes),
pch=c(as.character(params$clusters$index)),
col=params$clusters$index)
```

4.2 Interactive approach

For more control over the internal parameters used in anisotropy estimation, one needs to use the internal functions of intamap package. There are several options that the user can change. First of all, one can select the interpolation method used to construct the anisotropy estimation grid. Besides bilinear interpolation (akima package), bilinear and tps interpolation methods are also implemented and can be used interactively.

Below we give an example of a call to the **estimateAnisotropySc** function, and a list of the available arguments with a short description.

estimateAnisotropySc(x, y, r, len=length(x), method="linear",
min.x=min(x),
max.x=max(x), min.y=min(y), max.y=max(y), deb=FALSE, pl=FALSE,
br)

```
library(intamap)
data(meuse)

x=meuse$x
y=meuse$y
z=meuse$cadmium
```

Table B.2: Table of arguments used in the **estimateAnisotropySc** function

Input	Description
x	The x-axis coordinate of field
y	The y-axis coordinate of field
r	The field value in (xi,yi) point
len	default length(x), defines the rank of the number of cells to be used for the interpolation field. The total number of cells is defined as len * aspect, where aspect is the the aspect ratio (always > 1) of the coordinates
method	default "linear". This specifies the interpolation method used to construct the anisotropy estimation grid for derivatives estimation. Choices are: "cubic", "linear", "v4" (biharmonic spline)
min.x	default min(x), minimum x value for the interpolation grid
max.x	default max(x), maximum x value for the interpolation grid
min.y	default min(y), minimum y value for the interpolation grid
max.y	default max(y), maximum x value for the interpolation grid
deb	toggles debugging mode; only used interactively, outside intamap
pl	toggles plot; only used interactively, outside intamap
br	borders coordinates; only used interactively, outside intamap

```
# the default method - linear interpolation
result=intamap:::estimateAnisotropySc(x,y,z,pl=T)
print(result)

#plots the interpolated field and uses the biharmonic spline interpolation
#method.
result2=intamap:::estimateAnisotropySc(x,y,z,pl=T,method="v4")
print(result2)

#plots the interpolated field and uses the bicubic interpolation
#method.
result3=intamap:::estimateAnisotropySc(x,y,z,pl=T,method="cubic")
```

```
print(result3)

#define a 100 times bigger  interpolation grid
result4=intamap:::estimateAnisotropySc(x,y,z,len=length(x)*100,pl=T,
method="linear")
print(result4)
```

4.2.1 IntamapInteractive-Segmentation

Certain parameters can also be changed in the segmentation algorithm. The main internal function used for segmentation is **segmentData**.

Below we show an example of a call to **segmentData**, and a list of the available arguments with a short description.

segmentData=function(ddd,pl=FALSE,dev=FALSE,soft=0.2,br)

Table B.3: Table of arguments for the **segmentData** function

Input	Description
ddd	A nx2 or nx3 matrix containing the observations. First column is the horizontal and the second one the vertical coordinate.
pl	boolean variable that toggles plotting. If TRUE several plots during the segmentation procedure are shown. (Only used interactively)
dev	boolean variable that toggles saving the plots. (Only used interactively)
soft	parameter that controls the weight of sampling density and distance from neighbouring clusters in the cost function.
br	mx2 matrix with borders coordinates in case of plotting. This parameter is actually deactivated in line 79.
Output	
	A list with the following elements (i) index: a nx1 array with the indices. (ii) clusterNumber: The total number of clusters detected

```
library(intamapInteractive) #load intamap interactive package
```

```
data(quakes)            #load quakes dataset

x=quakes$lat
y=quakes$long
z=quakes$mag

xyz=cbind(x,y,z)

#default values use default values and enable plot
result=intamapInteractive:::segmentData(xyz,pl=T,soft=0.2)

#change weights for the final assignement
# you may also read the description of the doSegmentation function
#in order to get the details.
result=intamapInteractive:::segmentData(xyz,pl=T,soft=0.5)

#note that with less than 200 observation points, only one cluster will be
#returned.
```

4.2.2 IntamapInteractive-AnisotropyChoise

At this development stage it is not possible to pass other arguments besides the IntamapObject in the **anisotropyChoice** function. In order to get a different interpolation method or apply a different interpolation grid, someone will have to edit the source code of the **anisotropyChoice** function, using the knowledge gained from the previous paragraphs. We demonstrate below the parts of the code where these changes can be performed.

First, the parameter `soft` can be changed to a value different than 0.5 (in lines 78,79). Like the previous paragraph 4.2.1 one may use:

```
78:# do Segmentation
79:segmentResult=segmentData(ddd=xyz_d,pl=FALSE,dev=FALSE,soft=0.5)
```

One may also change the interpolation methods used to generate the anisotropy estimation grid (line 124) according to the examples in 4.2.

```
124:tempPar=intamap:::estimateAnisotropySc(temp[,1],temp[,2],temp[,3],
method="v4")
```

After the changes are completed, you can re-install the package source code with the command:

R CMD INSTALL intamapInteractive

In a Windows environment, you may want to consult this tutorial for the necessary tools required to build a package http://www.stat.osu.edu/~liuliang/research/R-package-windows.pdf, or you can use the **source** command to include the changes in your own scripts.

Appendix C

Help files for anisotropy functions in Intamap package

`estimateAnisotropy`
estimateAnisotropy

Description

This function estimates geometric anisotropy parameters for 2-D scattered data using the CTI method.

Usage

```
estimateAnisotropy(object,depVar)
```

Usage

```
estimateAnisotropy(object,depVar)
```

Arguments

`object`	(i) An Intamap type object (see `intamap-package`) containing one `SpatialPointsDataFrame` data frame named `observations` which includes the observed values (ii) or a `SpatialPointsDataFrame` which includes both coordinates and observations.

`depVar` name of the dependent variable; this is used only in case (ii).

Details

Given the input object that defines N coordinate pairs (x,y) and observed values (z), this method estimates of the geometric anisotropy parameters. Geometric anisotropy is a statistical property, which implies that the iso-level contours of the covariance function are elliptical. In this case the anisotropy is determined from the anisotropic ratio (R) and the orientation angle (*theta*) of the ellipse.

Assuming a Cartesian coordinate system of axes x and y, *theta* represents the angle between the horizontal axis and PA1, where PA1 is one of the principal axes of the ellipse, arbitrarily selected (PA2 will denote the other axis). R represents the ratio of the correlation along PA1 divided by the correlation length PA2. Note that the returned value of R is always greater than one (see `value` below.)

The estimation is based on the Covariance Tensor Identity (CTI) method. In CTI, the Hessian matrix of the covariance function is estimated from sample derivatives. The anisotropy parameters are estimated by explicit solutions of nonlinear equations that link (R,*theta*) with ratios of the covariance Hessian matrix elements.

To estimate the sample derivatives from scattered data, a background square lattice is used. The lattice extends in the horizontal direction from x.min to x.max where x.min (x.max) is equal to the minimum (maximum) x-coordinate of the data, and similarly in the vertical direction. The cell step in each direction is equal to the length of the lattice to the respective direction divided by the square root of N.

BiLinear interpolation, as implemented in `akima` package, is used to interpolate the field's z values at the nodes of the lattice.

The CTI method is described in detail in (Chorti and Hristopulos, 2008).

Note that to be compatible with `gstat` the returned estimate of the anisotropy ratio is always greater than 1.

Value

(i) If the input is an Intamap object, the value is a modification of the input object, containing a list element `anisPar` with the estimated anisotropy parameters. (ii)if the input is a `SpatialPointsDataFrame`,

then only the list `anisPar` is returned. The list `anisPar` contains the following elements:

`ratio` The estimate of the anisotropy ratio parameter. Using the degeneracy of the anisotropy under simultaneous ratio inversion and axis rotation transformations, the returned value of the ratio is always greater than 1.

`direction` The estimate of the anisotropy orientation angle. It returns the angle between the major anisotropy axis and the horizontal axis, and its value is in the interval (-90,90) degrees.

`Q` A 3x1 array containing the sample estimates of the diagonal and off-diagonal elements (Q11,Q22,Q12) of the covariance Hessian matrix evaluated at zero lag.

`doRotation` Boolean value indicating if the estimated anisotropy is statistically significant. This value is based on a statistical test of the isotropic (R= 1) hypothesis using a nonparametric approximation for the 95 percent confidence interval for R. This approximation leads to conservative (wider than the true) estimates of the confidence interval. If doRotation==TRUE then an isotropy restoring transformation (rotation and rescaling) is performed on the coordinates. If doRotation==FALSE no action is taken.

Note

This function uses `akima` package to perform "bilinear" interpolation. The source code also allows other interpolation methods, but this option is not available when the function is called from within INTAMAP.

In the `gstat` package, the anisotropy ratio is defined in the interval (0,1) and the orientation angle is the angle between the vertical axis and the major anisotropy axis, measured in the clockwise direction. If one wants to use ordinary kriging inside INTAMAP the necessary transformations are performed in the function `estimateParameters.automap`. If one wants to use ordinary kriging in the `gstat` package (but outside INTAMAP) the required transformations can be found in the source code of the `estimateParameters.automap` function.

Author(s)

A.Chorti, D.T.Hristopulos,G. Spiliopoulos

References

[1] http://www.intamap.org,

[2] A. Chorti and D. T. Hristopulos (2008). Non-parametric Identification of Anisotropic (Elliptic) Correlations in Spatially Distributed Data Sets, IEEE Transactions on Signal Processing, 56(10), 4738-4751 (2008).

[3] Em.Petrakis and D. T. Hristopulos (2009). A non-parametric test of statistical isotropy for Differentiable Spatial Random Fields in Two Dimensions. Work in progress. email: dionisi@mred.tuc.gr

Examples

```
library(intamap)
data(sic2004)
coordinates(sic.val)=~x+y
sic.val$value=sic.val$dayx

params=NULL

estimateAnisotropy(sic.val,depVar = "joker")
```

`rotateAnisotropicData`
rotateAnisotropicData

Description

This function applies an isotropic transformation of the coordinates specified in `object`.

Usage

```
rotateAnisotropicData(object,anisPar)
```

Usage

```
rotateAnisotropicData(object,anisPar)
```

Arguments

`object`	(i) An Intamap type object (see `intamap-package`) containing one `SpatialPointsDataFrame` data frame named `observations` which includes the observed values (ii) or a `SpatialPointsDataFrame` which includes both coordinates and observations or (iii) `SpatialPoints` which includes only coordinates to be rotated.
`anisPar`	An array containing the anisotropy parameters (anisotropy ratio and axes orientation) (see `estimateAnisotropy`) for the rotation. If `object` is the output of `estimateAnisotropy` function, these parameters are part of `object`. In cases (ii) and (iii) `anisPar` defines the two anisotropy parameters. For the definition of the anisotropy parameters see `estimateAnisotropy`.

Details

This function performs a rotation and rescaling of the coordinate axes in order to obtain a new coordinate system, in which the observations become statistically isotropic. This assumes that the estimates of the anisotropy ratio and the orientation angle provided in `anisPar` are accurate.

Value

(i) A modified object with transformed coordinates if rotateAnisotropic-Data is called with an Intamap object as input (see `intamap-package`) or (ii) the transformed coordinates if a `SpatialPointsDataFrame` is used as input or (iii) the transformed coordinates if a `SpatialPoints` object is the input.

Author(s)

Hristopulos Dionisis, Spiliopoulos Giannis

References

[1] http://www.intamap.org

[2] A. Chorti and D. T. Hristopulos (2008). Non-parametric Identification of Anisotropic (Elliptic) Correlations in Spatially Distributed Data Sets, IEEE Transactions on Signal Processing, 56(10), 4738-4751 (2008).

See Also

`estimateAnisotropy`

Examples

```
library(gstat)
data(sic2004)
coordinates(sic.val)=~x+y
sic.val$value=sic.val$dayx

params=NULL

obj<-list(
    observations=sic.val
    )
obj<-estimateAnisotropy(obj)
print(obj$anisPar)

obj$observations<-rotateAnisotropicData(obj$observations,obj$anisPar)

obj<-estimateAnisotropy(obj)
print(obj$anisPar)
```

`anisotropyChoice` *anisotropyChoice*

Description

This function combines segmentation of scattered 2D data and estimation of anisotropy parameters using the CTI method.

Usage

```
anisotropyChoice(object)
```

Usage

```
anisotropyChoice(object)
```

Arguments

`object`	An Intamap type object containing one `SpatialPointsDataFrame` with `observations`.

Details

The function `AnisotropyChoice` function employs the `doSegmentation` function to automatically separate the original dataset into clusters based on the sampling density and the spatial locations of the data (see `doSegmentation` for details). The results of the segmentation procedure and the anisotropy analysis per cluster are returned in a matrix of dimension [cl]x5, where [cl] is the number of clusters . Each row of the matrix contains the cluster index, the anisotropy ratio, the anisotropy direction, the number of cluster points and the area inside the convex hull of the cluster. In addition, a single set of anisotropy parameters is returned in the element `anisPar`. These parameters are calculated using weighted averages of the covariance Hessian matrix estimates in each cluster. The weights are based on the area enclosed by the convex hull of each cluster.

Value

`object`: A modified Intamap type object is returned, which contains the results of the anisotropy parameter estimation. The anisotropy parameters are returned in the element `anisPar` as described below.

`anisPar` List element in `object` that contains a list with the following elements:

- `ratio`A coarse-grained anisotropy ratio for all the data
- `direction`A coarse-grained anisotropy orientation for all the data
- `clusters`A matrix of dimension [cl]x5 which determines the anisotropy per cluster. Each row of `clusters` gives the (cluster id, anisotropy ratio, anisotropy direction, number of points, area) for each cluster detected.

`clusters` list element added to the original object containing the segmentation results.

- indexIndex array identifying the cluster in which each observation point belongs. Zero value means that the observations has been removed.
- clusterNumberNumber of clusters detected.

Note

This function uses the `akima` package to perform "bilinear" and "bicubic" interpolation for the estimation of spatial derivatives

Author(s)

D.T. Hristopulos, G.Spiliopoulos, A.Chorti

References

[1] http://www.intamap.org

[2] A. Chorti and D. T. Hristopulos (2008). Non-parametric Identification of Anisotropic (Elliptic) Correlations in Spatially Distributed Data Sets, IEEE Transactions on Signal Processing, 56(10), 4738-4751 (2008).

[3] D. T. Hristopulos, M. P. Petrakis, G. Spiliopoulos, A. Chorti (2009). Non-parametric estimation of geometric anisotropy from environmental

sensor network measurements, StatGIS 2009: Geoinformatics for Environmental Surveillance Proceedings (ed. G. Dubois).

Examples

```
library(intamapInteractive)

data(walker)
coordinates(walker)=~X+Y
object=createIntamapObject(observations=walker)
object=anisotropyChoice(object)

print(summary(object$clusters$index))
print(object$anisPar)
```

`doSegmentation` *Spatial Segmentation - Clustering for Scattered Observations*

Description

This function performs segmentation of scattered 2D data based on sampling density and location.

Usage

```
doSegmentation(object)
```

Usage

```
doSegmentation(object)
```

Arguments

`object`	An Intamap type object containing the element (list) `observations`, which includes the coordinates of the observation locations

Details

This function performs segmentation of scattered 2D data based on sampling density and location. Let us assume that `No` is the number of observation locations. If `No¡` 200, then a single cluster is returned. (1) The segmentation algorithm first removes isolated distant points, if there are any, from the observation locations. Points (xi, yi) are characterized as 'isolated' and 'distant' if they satisfy the following conditions : $abs(xi - mean(x)) > 4 * std(x) or abs(yi - mean(y)) > 4 * std(y)$ and distance from closest neighbor $> \backslash sqrt(std(x)/2)^2 + (std(y)/2)^2$. After the first step the size of the original dataset is reduced to N (N= No - isolated points) points. (2) A sampling density matrix (lattice) consisting of N cells that cover the study area is constructed. Each cell is assigned a density value equal to the number of observation points inside the cell. In addition, each observation point is assigned the sampling density value of the containing cell. (3) Unsupervised clustering edge detection is used to determine potential cluster perimeters. (4) Each closed region's perimeter is labeled with a different cluster (segment) number. (5) All observation

points internal to a cluster perimeter are assigned to the specific cluster. (6) Each cluster that contains fewer than 50 observation points is rejected. (7) The observation points that have not initially been assigned to a cluster and those belonging to rejected (small) clusters are assigned at this stage. The assignment takes into account both the distance of the points from the centroids of the accepted clusters as well as the mean sampling density of the clusters.

Note: The `No¡` 200 empirical constraint is used to avoid extreme situations in which the sampling density is concentrated inside a few cells of the background lattice, thereby inhibiting the edge detection algorithm.

Value

A modified Intamap object which additionally includes the list element `clusters`. This element is a list that contains (i) the indices of removed points from `observations`; (ii) the indices of the clusters to which the remaining observation points are assigned and (iii) the number of clusters detected.

`clusters` list element added to the original object containing the segmentation results.

- rmdistIndices of removed points.
- indexIndex array identifying the cluster in which each observation point belongs.
- clusterNumberNumber of clusters detected.

Author(s)

A. Chorti, Spiliopoulos Giannis, Hristopulos Dionisis

References

[1] D. T. Hristopulos, M. P. Petrakis, G. Spiliopoulos, A. Chorti (2009). Non-parametric estimation of geometric anisotropy from environmental sensor network measurements, StatGIS 2009: Geoinformatics for Environmental Surveillance Proceedings (ed. G. Dubois).

Examples

```
library(intamapInteractive)
```

```
data(walker)
coordinates(walker)=~X+Y
object=createIntamapObject(observations=walker)
object=doSegmentation(object)

print(summary(object$clusters$index))
```

Bibliography

[Abr97] P. Abrahamsen. A review of Gaussian random fields and correlation functions - second edition. Technical Report 917, Norwegian Computing Center, Oslo, Norway, Box 114, Blindern, N-0314, 1997.

[Adl81] R. Adler. *The Geometry of Random Field Models*. John Wiley & Sons, New York, NY, USA, 1981.

[AGGR05] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data. *Data Mining and Knowledge Discovery*, 11(1):5–33, 2005.

[BAM03] R. Bammer, B. Acar, and M.E. Moseley. In vivo MR tractography using diffusion imaging. *European journal of radiology*, 45(3):223, 2003.

[Bea02] C. Beaulieu. The basis of anisotropic water diffusion in the nervous system. *NMR in Biomedicine*, 15(7-8):435–455, 2002.

[Bes74] J. Besag. Spatial interaction and statistical analysis of lattice systems. *Journal of the Royal Statistical Society*, 36(2):192–236, 1974.

[BP96] P.J. Basser and C. Pierpaoli. Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor mri. *Journal of Magnetic Resonance-Series B*, 111(3):209–219, 1996.

[CCHR76] GG Cleveland, DC Chang, CF Hazlewood, and HE Rorschach. Nuclear magnetic resonance measurement of skeletal muscle: anisotrophy of the diffusion coefficient of the intracellular water. *Biophysical Journal*, 16(9):1043–1053, 1976.

[CFZ99] Chun-Hung Cheng, Ada Waichee Fu, and Yi Zhang. Entropy-based subspace clustering for mining numerical data. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on knowledge discovery and data mining*, pages 84–93, 1999.

[CH08] A. Chorti and D. T. Hristopulos. Non-parametric identification of anisotropic (elliptic) correlations in spatially distributed data sets. *IEEE Transactions on Signal Processing*, 56(10):4738–4751, 2008.

[Cla79] I. Clark. *Practical Geostatistics*. Applied Science Publishers, Essex, England, 1979.

[Cre90] N. Cressie. The origins of kriging. *Mathematical Geology*, 22(3):239–253, 1990.

[CST00] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, Cambridge, 2000.

[Dam71] R. Damadian. Tumor detection by NMR. *Science*, 171:1151–1153, 1971.

[DL08] Joan Dawson and Paul C. Lauterbur. Magnetic resonance imaging. *Scholarpedia*, 2008.

[Doo53] J.L. Doob. *Stohastic Processes*, page 654. John Willey & Sons, New York, 1953.

[DP00] Andrew Moore Dan Pelleg. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 727–734, San Francisco, 2000. Morgan Kaufmann.

[DTP+91] P. Douek, R. Turner, J. Pekar, N. Patronas, and D.L. Bihan. MR color mapping of myelin fiber orientation. *Journal of computer assisted tomography*, 15(6):923, 1991.

[EG99] M. D. Ecker and A.E. Gelfand. Bayesian modeling and inference for geometrically anisotropic spatial data. *Mathematical Geology*, 32(1):67–82, 1999.

[Ehr97] J. Ehrhardt. The RODOS system: Decision support for off-site emergency management in Europe. *Radiation Protection Dosimetry*, 73(1-4):35–40, 1997.

[EKSX96] M. Ester, H.P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. KDD*, volume 96, pages 226–231, 1996.

[Gan63] L. Gandin. Objective analysis of meteorological fields, gridromet, leningrad. *English tranlation, Israel Program for Scientific Translation, Jerusalem*, 1963.

[GMW07] G. Gan, C. Ma, and J. Wu. *Data Clustering Theory, Algorithms and Applications*. SIAM, Philadelphia, USA, 2007.

[GW06] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[HE07] D. T. Hristopulos and S. Elogne. Analytic properties and covariance functions of a new class of generalized Gibbs random fields. *IEEE Transactions on Information Theory*, 53(12):4667–4679, 2007.

[HMAB07] D. T. Hristopulos, S. Mertikas, I. Arhontakis, and J. Brownjohn. Using GPS for monitoring tall-building response to wind loading: filtering of abrupt changes and low-frequency noise, variography and spectral analysis of displacements. *GPS Solutions*, 11(2):85–95, 2007.

[Hoh91] M. E. Hohn. An introduction to applied geostatistics. by edward h. isaaks and r. mohan srivastava, 1989, oxford university press, new york, 561 p., isbn 0-19-505012-6, isbn 0-19-505013-4 (paperback). *Computers and Geosciences*, 17(3):471–473, 1991.

[HPTH09] P.H. Hiemstra, E.J. Pebesma, C.J.W. Twenhöfel, and G.B.M. Heuvelink. Real-time automatic interpolation of ambient gamma dose rates from the Dutch radioactivity monitoring network. *Computers and Geosciences*, 35(8):1711–1721, 2009.

[Hri02] D. T. Hristopulos. New anisotropic covariance models and estimation of anisotropic parameters based on the covariance tensor identity. *Stochastic Environmental Research and Risk Assessment*, 16(1):43–62, 2002.

[Hri03] D. T. Hristopulos. Permissibility of fractal exponents and models of band-limited two-point functions for fGn and fBm random fields. *Stochastic Environmental Research and Risk Assessment*, 17(3):191–216, 2003.

[HS92] Robert M. Haralick and Linda G. Shapiro. *Computer and Robot Vision*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1992.

[INS09] Infrastructure for Spatial Information in the European Community. EU 7th Framework Programme, 2009.

[Kit83] P. K. Kitanidis. Statistical estimation of polynomial generalized covariance functions and hydrologic applications. *Water Resources Res.*, 19(2):909–921, 1983.

[Kit87] P. K. Kitanidis. Parametric estimation of covariances of regionalized variables. *Water Resources Res.*, 23(4):671–680, 1987.

[KKK04] Karin Kailing, Hans-Peter Kriegel, and Peer Kröger. Density-connected subspace clustering for high-dimensional data. In *Proceedings 4th SIAM International Conference on Data Mining, Lake Buena Vista, FL, USA*, pages 246–257, 2004.

[KVCS96] SP Kałużny, SC Vega, TP Cardoso, and AA Shelly. S+ SPATIALSTATS–Users Manual. *MathSoft Inc., Seattle*, 1996.

[KW95] R.E. Kass and L. Wasserman. A reference bayesian test for nested hypotheses and its relationship to the schwarz criterion. *Journal of the American Statistical Association*, 90(431):928–934, 1995.

[Mat63] G. Matheron. Principles of geostatistics. *Economic geology*, 58(8):1246, 1963.

[MCK+90] ME Moseley, Y. Cohen, J. Kucharczyk, J. Mintorovitch, HS Asgari, MF Wendland, J. Tsuruda, and D. Norman. Diffusion-weighted MR imaging of anisotropic water diffusion in cat central nervous system. *Radiology*, 176(2):439, 1990.

[MKAN91] ME Moseley, J. Kucharczyk, HS Asgari, and D. Norman. Anisotropy in diffusion-weighted MRI. *Magnetic resonance in medicine: official journal of the Society of Magnetic Resonance in Medicine/Society of Magnetic Resonance in Medicine*, 19(2):321, 1991.

[MP00] G. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley & Sons, New York, USA, 2000.

[Peb04] Edzer J. Pebesma. Multivariable geostatistics in S: the gstat package. *Computers and Geosciences*, 30:683–691, 2004.

[PH10] M. P. Petrakis and D. T. Hristopulos. On the joint probability density function of geometric anisotropy statistics for two dimensional differentiable random fields and a non-parametric test of statistical isotropy. Working Draft, 2010.

[PI98] E. Pardo-Igúzquiza. Maximum likelihhod estimation of spatial covariance parameters. *Mathematical Geology*, 30(1):95–108, 1998.

[RDD73] P. Hart R. Duda and D.Stork. *Pattern Classification*. Wiley, 2nd edition, 1973.

[SB07] U. Stoehlker and M. Bleher. Early warning systems: Identifying anomalies with monitoring systems. INTAMAP (6th FP, ICT for Environmental Risk Management) project report D5.1, 2007.

[SHPC] I. Spiliopoulos, D.T. Hristopulos, M.P. Petrakis, and A. Chorti. A multigrid method for the estimation of geometric anisotropy in enviromental data from sensor networks. *Computers and Geosiences*. In Press accepted 2010.

[SIC] http://www.ai-geostats.org/events/sic2004/index.htm.

[Swe62] P. Swerling. Statistical properties of the contours of random surfaces. *IRE Transactions on Information Theory*, 8:315–321, 1962.

[USB09] M. Bleher U. Stoehlker and S. Burbeck. Generating datasets for routine and emergency monitoring. INTAMAP (6th FP, ICT for Environmental Risk Management) project report D5.4, 2009.

[Van88] E. VanMarcke. *Random Fields*, page 31. MIT Press, 3rd edition, 1988.

[vGdVD+94] P. van Gelderen, M.H.M. de Vleeschouwer, D. DesPres, J. Pekar, P.C.M. van Zijl, and C.T.W. Moonen. Water diffusion

and acute stroke. *Magnetic Resonance in Medicine*, 31(2):154–163, 1994.

[Wac03] H. Wackernagel. *Multivariate Geostatistics*. Springer Verlag, Berlin, 2003.

[Yag62] AM Yaglom. An Introduction to the Theory of Stationary Random Fields. 235 pp. *Prentice-Hall, Englewood Cliffs, NJ*, 1962.

[Zim93] D. L. Zimmerman. Another look at anisotropy in geostatistics. *Mathematical Geology*, 25(4):453–470, 1993.