

# Gamma-Ray Astronomy Data Analysis Framework based on the Quantification of Background Morphologies using Minkowski Tensors

Methoden zur Datenanalyse in der Gammaastronomie  
basierend auf der Quantifizierung der Morphologie des  
Untergrunds mit Hilfe der Minkowski-Tensoren

Der Naturwissenschaftlichen Fakultät der  
Friedrich-Alexander-Universität Erlangen-Nürnberg

zur

Erlangung des Doktorgrades Dr. rer. nat.

vorgelegt von  
Daniel Göring  
aus Nürnberg



Als Dissertation genehmigt von der Naturwissenschaftlichen Fakultät der  
Friedrich-Alexander Universität Erlangen-Nürnberg

Tag der mündlichen Prüfung: 29. Oktober 2012

Vorsitzender der	
Promotionskommission:	Prof. Dr. Rainer Fink
Erstberichterstatte:	Prof. Dr. Christian Stegmann
Zweitberichterstatte:	Prof. Dr. Klaus Mecke
Drittberichterstatte:	Prof. Dr. Thomas Lohse

# Abstract

The aim of this work was to incorporate structure information into the analysis of very high energy (VHE) gamma-ray astronomy data. Usually analysis frameworks in VHE gamma-ray astronomy are based exclusively on the local excess of gamma-like event counts compared to the expected count of background events. The morphology and spatial structure of the corresponding gamma-ray counts map are not used in these frameworks. Considering the steadily increasing number of extended sources detected in VHE gamma-rays and the rapid progress of this young branch of astronomy, a refined analysis based on the spatial structure of the gamma-ray counts map promises an increased sensitivity to extended sources and a better understanding of observational data.

The spatial structure of gamma-ray counts maps can be quantified using Minkowski tensors. These tools from integral geometry provide a powerful basis for the morphometric analysis of counts maps. While still rather unknown in the gamma-ray astronomy community, the Minkowski tensors were already very successfully applied to a wide range of different problems in statistical physics.

In this work an algorithm to compute the exact multivariate probability distribution of an arbitrary set of Minkowski tensors applied to a gamma-ray counts map was derived. Using these distributions one is able to quantify the significance of a measured structure compared to the typical structure of a background measurement. Results of analyses based on the significance of local structure deviations presented in this thesis proved to be in good agreement with well established analyses.

The quantification of local structure deviations is based exclusively on a characterization of typical background structures and therefore no a priori models of potential gamma-ray sources are needed, as opposed to e.g. fitting a full likelihood model to the data. While precomputing the required probability distributions for a structure analysis is very time-consuming, the actual structure analysis turned out to be very fast. Currently the observable



---

sensitivity gain of a structure analysis compared to an analysis based on excess counts is limited by the available probability distributions. Further studies on future computers are needed to fully characterize the sensitivity of the developed framework. Based on the complexity of the algorithm and the current progress in computer hardware, the author expects the required distributions to be available within the next few years.

# Zusammenfassung

Ziel dieser Arbeit war es, die Datenanalyse in der Gammaastronomie bei höchsten Energien um die Auswertung von Strukturinformationen zu erweitern. Üblicherweise beschränken sich die Analysemethoden in der Gammaastronomie bei höchsten Energien darauf, den Überschuss gammaartiger Ereignisse bezüglich der erwarteten Zahl an Untergrundereignissen zu quantifizieren. Die Morphologie und räumliche Struktur der entsprechenden Himmelskarten werden in diesen Analysen nicht berücksichtigt. In Anbetracht der kontinuierlich wachsenden Zahl an entdeckten ausgedehnten Quellen höchst-energetischer Gammastrahlung und des rasanten Fortschritts dieses jungen Feldes der Astronomie verspricht eine überarbeitete Analyse basierend auf der räumlichen Struktur der Gammastrahlungs-Himmelskarten eine höhere Sensitivität für die Detektion ausgedehnter Quellen und ein besseres Verständnis der Beobachtungen.

Die räumliche Struktur der Gammastrahlungs-Himmelskarten kann mit Hilfe der Minkowski-Tensoren quantifiziert werden. Diese Werkzeuge aus der Integralgeometrie bilden eine leistungsfähige Grundlage für die morphometrische Analyse der Himmelskarten. Während sie in der Gammaastronomie noch weitgehend unbekannt sind, wurden die Minkowski-Tensoren in der statistischen Physik bereits überaus erfolgreich auf eine Vielzahl verschiedener Probleme angewendet.

In dieser Arbeit wurde ein Algorithmus hergeleitet, mit dessen Hilfe sich die exakten Wahrscheinlichkeitsverteilungen beliebiger Minkowski-Tensoren für Himmelskarten berechnen lassen. Durch diese Verteilungen ist es möglich, die Signifikanz einer gemessenen Struktur im Vergleich zur typischen Struktur einer Untergrundmessung zu bestimmen. Die in dieser Dissertation vorgestellten Ergebnisse der Analysen basierend auf der Signifikanz solcher Strukturabweichungen zeigten eine gute Übereinstimmung mit den Ergebnissen etablierter Analysemethoden.

Die Analyse lokaler Strukturabweichungen beruht ausschließlich auf der Charakterisierung typischer Untergrundstrukturen und es sind keine a pri-

---

ori Annahmen über mögliche Gammastrahlungsquellen in den Himmelskarten nötig, welche z.B. für den Fit eines vollständigen Likelihood-Modells unumgänglich sind. Während die vorausgehende Berechnung der für die Strukturanalyse nötigen Wahrscheinlichkeitsverteilungen sehr aufwändig und langwierig ist, stellte sich heraus, dass die eigentliche Strukturanalyse sehr schnell durchgeführt werden kann. Zur Zeit sind die Verbesserungen der Strukturanalyse gegenüber einer Analyse basierend auf dem Überschuss an gemessenen Ereignissen limitiert durch die verfügbaren Wahrscheinlichkeitsverteilungen. Weitere Untersuchungen auf künftigen Computern sind nötig, um die Eigenschaften der entwickelten Analyse vollständig zu charakterisieren. Ausgehend von der Komplexität des Algorithmus und dem gegenwärtigen Fortschritt der Computertechnik erwartet der Autor, dass die nötigen Verteilungen innerhalb der nächsten Jahre zugänglich werden.

# Contents

<b>1</b>	<b>Introduction</b>	<b>10</b>
<b>2</b>	<b>Minkowski Tensors</b>	<b>16</b>
2.1	Minkowski Functionals and Hadwiger's theorem . . . . .	16
2.1.1	General Properties . . . . .	16
2.1.2	Geometric Interpretation . . . . .	19
2.1.3	Summary . . . . .	21
2.1.4	Functionals for Pixel Images . . . . .	22
2.2	Minkowski Tensors and Alesker's theorem . . . . .	32
2.2.1	General Properties . . . . .	32
2.2.2	Summary . . . . .	38
2.2.3	Tensors for Pixel Images . . . . .	38
<b>3</b>	<b>Computing Distributions of Minkowski Tensors</b>	<b>49</b>
3.1	Partition Numbers . . . . .	49
3.2	Distributions of the Area Tensors . . . . .	51
3.3	Distributions of Minkowski Tensors . . . . .	54
3.3.1	Contribution of Interior Pixels . . . . .	55
3.3.2	Contribution of Boundary Pixels . . . . .	59
3.3.3	Construction of the Probability Distribution . . . . .	74
3.4	Comparison to Brute Force Approach . . . . .	76
3.4.1	Complexity Analysis . . . . .	76
3.4.2	Benchmark Results . . . . .	78
<b>4</b>	<b>Minkowski Maps</b>	<b>85</b>
4.1	Quantification of Structure Deviations . . . . .	85
4.2	Local Structure Analysis . . . . .	90
<b>5</b>	<b>Background Correction</b>	<b>97</b>
5.1	Exposure Reduction via Post-Selection . . . . .	98

5.2	Exposure Gain via Monte Carlo Observations . . . . .	99
5.3	Robust Fit of Background Models . . . . .	101
<b>6</b>	<b>Application to Gamma-Ray Astronomy Data</b>	<b>109</b>
6.1	The H.E.S.S. Experiment . . . . .	109
6.1.1	Background Model . . . . .	110
6.1.2	The Crab Nebula, a View of the Standard Candle . .	113
6.1.3	RX J1713.7–3946, a View of an Extended Source . .	120
6.1.4	The Galactic Scan, a View of a Large Scale Analysis .	124
6.2	The Fermi Satellite Mission . . . . .	128
6.2.1	Background Model . . . . .	128
6.2.2	PKS 2155–304, a View of an Extragalactic Point Source	130
6.2.3	RX J1713.7–3946, a View of a Galactic Source . . . .	133
<b>7</b>	<b>Systematic Studies</b>	<b>137</b>
7.1	Analysis Run-Time . . . . .	137
7.2	Influence of Scan Window Size . . . . .	138
7.3	Influence of Background Correction . . . . .	143
7.4	Sensitivity of Different Tensors . . . . .	144
7.5	Influence of Boundary Conditions . . . . .	150
7.6	Influence of Bin Size . . . . .	152
<b>8</b>	<b>Conclusion</b>	<b>154</b>
8.1	Summary . . . . .	154
8.2	Outlook . . . . .	155
	<b>List of Figures</b>	<b>157</b>
	<b>List of Tables</b>	<b>160</b>
	<b>Bibliography</b>	<b>161</b>

# 1 Introduction

Ground-based very high energy (VHE) gamma-ray astrophysics took a flying leap forward in the past few years (e.g. Buckley et al., 2008). Starting from just a few hardly detectable point sources the field evolved into a rich branch of astronomy comprising morphological studies of extended sources and diffuse gamma rays from the galactic center region and modeling of supernova remnants (SNR) and pulsar wind nebulae (PWN) — to name but a few. Even in the light of this progress, there are still many open questions and future experiments like the Cherenkov Telescope Array (CTA) promise exciting new insights in the physics of VHE cosmic ray accelerators (Actis et al., 2011).

The amazing progress of ground-based gamma-ray astronomy is not solely owed to technical advances and the use of novel detectors but also to deeper understanding of the experiments. Using detailed models and Monte Carlo simulations the event reconstruction of the High Energy Stereoscopic System (H.E.S.S.) — a ground-based gamma-ray telescope array located in the Khomas Highland of Namibia (F. Aharonian et al., 2006) (cf. figure 1.1) — was improved substantially. This led to an increase of the angular and energy resolution and a superior background rejection compared to earlier techniques. The new reconstruction almost doubled the sensitivity of the telescope array (de Naurois and Rolland, 2009).

While the analysis tool chains used by the different experiments to process the reconstructed events are without a doubt very powerful and received lots of fine tuning and improvements over the years, they are still mainly based on the likelihood ratio test of Li and Ma (1983). This test was perfectly suited for the early days of VHE gamma-ray astronomy, when only a few point sources were known and nobody dared to dream of the amount and diversity of sources known today. However, the test is exclusively based on the overall amount of excess photons compared to the expected amount of background photons and thus can not take into account the spatial distribution of the photons. In the analysis of point-like sources this does not pose a problem, as point sources



Figure 1.1: The four H.E.S.S. telescopes in the Khomas Highland of Namibia  
Image credit: H.E.S.S. Collaboration

show no spatial structure except for excess photon counts at the position of the source on top of the expected background photons. Nevertheless, the currently known VHE gamma-ray source population contains large scale structures like the diffuse emission from the galactic center ridge (Aharonian et al., 2006) and many extended sources (cf. figure 1.2). Using this additional information about the spatial structure of gamma-ray counts maps in a specialized analysis might improve the sensitivity of current experiments considerably.

Building an analysis based on the spatial structure of the detected gamma-ray intensity distribution can improve the sensitivity to extended structures and might help to detect faint extended sources, too weak to be seen with analyses based on Li and Ma (1983). Such a morphometric analysis can be very helpful in studying potential large scale sources like diffuse emission along the galactic plane or emission from the “Fermi Bubbles” shown in figure 1.3, which could not be detected in the TeV energy band so far.

In order to use the spatial structure of a gamma-ray counts map in the analysis of the reconstructed events, you need to find a way to quantify structure. Such a way is provided by the mathematical discipline of integral geometry via the so called Minkowski tensors. These tensors are a simple<sup>1</sup> yet

---

<sup>1</sup>A detailed discussion follows in later sections, but to give a rough idea of the meaning of *simple* in this context: Due to their additivity (as well as motion covariance and convex continuity) the Minkowski tensors are like the “linear functions” of structure functionals.

## 1 Introduction

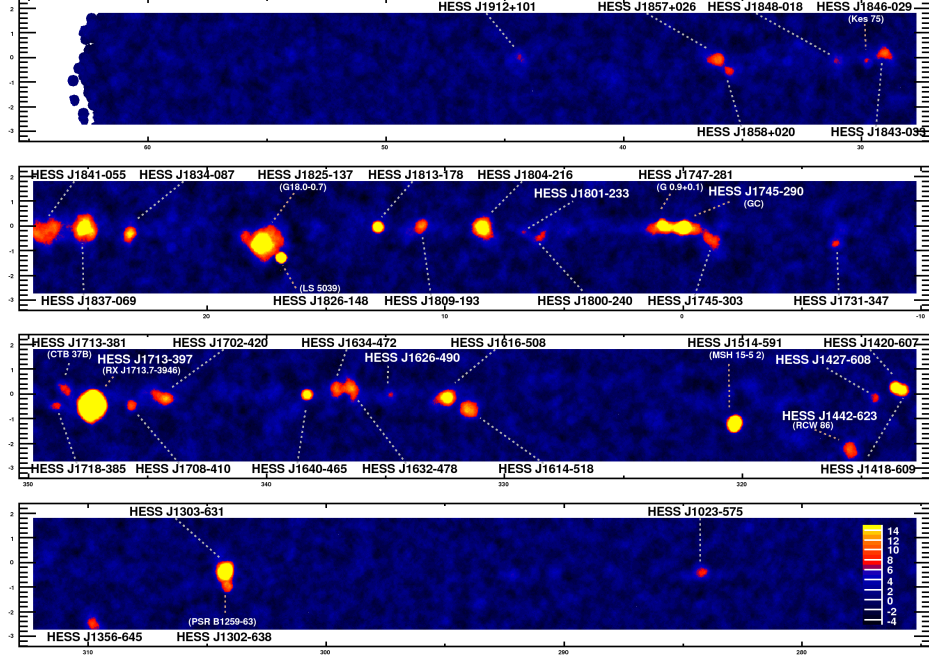


Figure 1.2: Galactic plane as seen by H.E.S.S. in 2007. The color indicates the significance of the observed signal in each bin and the position is given in galactic coordinates.

Image credit: H.E.S.S. Collaboration

powerful<sup>2</sup> tool to quantify structures and study morphologies. They have very successfully been applied to various problems of statistical physics (e.g. Mecke, 1994, 1998, 2000; Schröder-Turk et al., 2010, 2011). While the Minkowski tensors are not yet very commonly known among astrophysicists, they were already used to study problems like the distribution of galaxy clusters (Mecke et al., 1994) or the anisotropy of the cosmic microwave background (Schmalzing and Górski, 1998). This thesis shows how to apply the Minkowski tensors to gamma-ray counts maps and how to use them in order detect VHE gamma-ray

<sup>2</sup>Their “nice” properties make it feasible to use them in complex derivations and there are theorems proving the completeness of this set of morphometric functionals. Details are given in later sections.



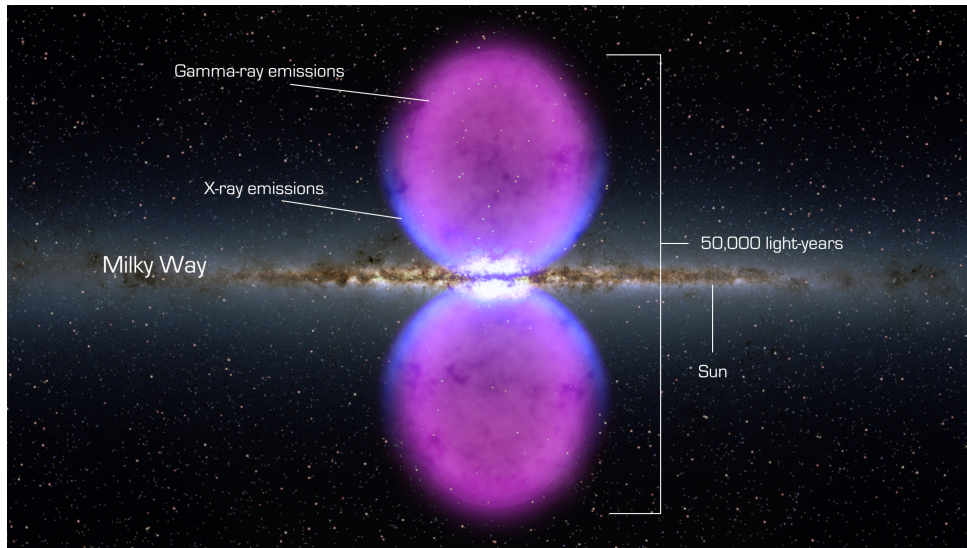


Figure 1.3: Artistic view of the “Fermi Bubbles”, large regions of GeV gamma ray emission perpendicular to the galactic plane (Dobler et al., 2010; Su et al., 2010)

Image credit: NASA/GSFC

sources.

The main idea of this morphometric analysis is to understand the typical structure of a background measurement. Gamma-ray sources can then be identified by looking for significant deviations from this expected background morphology. In order to quantify the typical structure of a pure background measurement, you need to pick a background model. It is reasonable to assume a Poisson statistic for the background events, just like Li and Ma (1983) do. The additional spatial model used in this thesis is just a homogeneous and isotropic distribution of the background events across the observed sky. This assumption is motivated by the fact that the most common source of background events in ground-based VHE gamma-ray astronomy are charged cosmic rays, which cause electromagnetic showers in Earth’s atmosphere, very similar to the showers produced by VHE gamma rays. Thus, some charged cosmic rays are indistinguishable from VHE gamma rays for ground-based

experiments, which have to reconstruct the incident photons by observing the induced electromagnetic showers.<sup>3</sup> While these charged cosmic rays are expected to originate from similar sources as the observed VHE gamma rays, they lose their direction correlation with their origin in interstellar magnetic fields and hit Earth from randomized directions. This leads to a quite homogeneous and isotropic flux of background events. Using an equal area binning for the creation of gamma-ray counts maps, these assumptions lead to the following background model:

$$\mathcal{P}(n|i, j) = \frac{\lambda^n}{n!} e^{-\lambda}, \quad (1.1)$$

i.e. the probability to detect  $n$  gamma-like events in the bin  $(i, j)$  of the sky map is just given by the corresponding Poisson probability. Due to the homogeneity and isotropy of the background model, this probability does not depend on  $(i, j)$  and the model is fully determined by the only free parameter  $\lambda$ , which describes the expected background level.

Based on equation (1.1) it is possible to derive the corresponding probability distributions of a given set of Minkowski tensors. By comparing the values of the Minkowski tensors of a measured gamma-ray counts map to this probability distribution you can test the hypothesis that the given measurement shows only background fluctuations, which allows you to identify additional structures on top of the expected background.

In chapter 2 the Minkowski tensors are introduced, a short overview of their most important properties is given and it is shown how to use them to quantify structures. Afterwards an efficient algorithm to compute probability distributions of Minkowski tensors for the background model described by equation (1.1) is derived in chapter 3. Chapter 4 shows how to quantify deviations from expected background structures using these distributions and how to visualize the deviations in Minkowski sky maps. After establishing this solid theoretical basis of the structure analysis, the following chapters demonstrate how to apply the techniques to real data. Within this discussion chapters 5 shows how to correct for detector effects and other influences which might distort the homogeneity and isotropy of the observed background events.

---

<sup>3</sup>Our atmosphere is not opaque to gamma rays, so they can only be studied directly by space-based experiments like satellites. Ground-based telescopes have to study secondary particles like those of the induced electromagnetic shower or the Cherenkov light emitted by highly relativistic particles in this shower.

---

In order to demonstrate the capabilities of the morphometric analysis and the validity of the used assumptions and methods, the structure analysis is applied to data of the H.E.S.S. experiment and compared to the results of the standard analysis tool chain in chapter 6. In a short digression it is shown that the developed techniques can even be used to analyze data of satellite experiments by applying it to data taken with the Large Area Telescope (LAT) on the Fermi gamma-ray satellite (Atwood et al., 2009). Finally the characteristics of this new analysis are studied in chapter 7 using detailed Monte Carlo simulations and a summary of the most important results is given in chapter 8.

## 2 Minkowski Tensors

Minkowski tensors are a natural generalization of the scalar Minkowski functionals,<sup>1</sup> which in turn are the Minkowski tensors of rank 0. The next section provides an introduction to structure quantification by describing the Minkowski functionals and discussing their properties. In section 2.2 these ideas are generalized to the concept of Minkowski tensors.

### 2.1 Minkowski Functionals and Hadwiger's theorem

Originally the Minkowski functionals were introduced to study compact convex sets. This restriction is still reflected in one of their defining properties, but — as shown later in this section — the domain of the Minkowski functionals can immediately be generalized to polyconvex sets, i.e. finite unions of compact convex sets. Being confined to the study of polyconvex bodies is usually no restriction at all from a physicist's point of view. Most experimental results are inherently binned due to finite experimental resolution and quantization errors. Studying binned — i.e. pixelated — images is equivalent to studying polyconvex bodies, as all image structures are finite unions of pixels, which in turn are compact convex sets.

#### 2.1.1 General Properties

In order to examine structure functionals on  $\mathcal{K}$ , the set of closed, compact and convex bodies, it is useful to note that  $\mathcal{K}$  is closed under intersections. So,  $A, B \in \mathcal{K} \Rightarrow A \cap B \in \mathcal{K}$ , i.e.  $A \cap B$  is closed, compact and convex.<sup>2</sup> The

---

<sup>1</sup>The scalar Minkowski functionals are also known as quermassintegrals, intrinsic volumes or curvature integrals.

<sup>2</sup>*Closed*:  $A, B$  closed  $\Leftrightarrow A^c, B^c$  open;  $A \cap B = (A^c \cup B^c)^c$ ,  $A^c \cup B^c$  open (union of open sets)  $\Rightarrow A \cap B$  closed. *Compact*: Any closed subset  $X$  of the compact set  $A$  is compact, as every open cover of  $X$  can be extended to an open cover of  $A$  by adding  $X^c$  to the cover. For this cover of  $A$  a finite subcover of  $A$  exists, as  $A$  is compact. Removing  $X^c$  from this finite

set of polyconvex bodies  $\mathcal{R}$  is by definition closed under unions and inherits the property to be closed under intersections from  $\mathcal{K}$ .<sup>3</sup> Now let  $M : \mathcal{K} \rightarrow \mathbb{R}$  be a functional on the space of convex bodies with the following properties:

**Additivity**

$$M(A \cup B) = M(A) + M(B) - M(A \cap B). \quad (2.1)$$

**Motion invariance**

$$M(gA) = M(A) \quad \forall g \in \mathcal{G}, \quad (2.2)$$

where  $\mathcal{G}$  denotes the group of motions, i.e. translations and rotations in  $\mathbb{R}^d$ .

**Continuity**

$$\lim_{n \rightarrow \infty} A_n = A \Rightarrow \lim_{n \rightarrow \infty} M(A_n) = M(A). \quad (2.3)$$

A remarkable theorem by Hadwiger states that in  $d$  dimensions

$$M(A) = \sum_{\nu=0}^d c_\nu M_\nu(A), \quad (2.4)$$

i.e. every additive, motion-invariant and continuous functional  $M : \mathcal{K} \rightarrow \mathbb{R}$  is a linear combination of the  $d+1$  Minkowski functionals  $M_\nu$  (Hadwiger, 1957). One further property makes the Minkowski functionals (up to a multiplicative constant) a unique choice over other sets of  $d+1$  independent functionals:

$$M_\nu(\lambda A) = \lambda^{d-\nu} M_\nu(A), \quad (2.5)$$

i.e. the Minkowski functionals are homogeneous functions of order  $d-\nu$  under scaling of the convex body by a constant  $\lambda \in \mathbb{R}$ . This assures that

---

subcover yields a finite subcover of the initial cover of  $X$ .  $\Rightarrow A \cap B$  compact. *Convex*:  $A$  convex  $\Leftrightarrow \overline{xy} \subset A \quad \forall x, y \in A$ , where  $\overline{xy}$  is the line segment connecting  $x$  and  $y$ ; analogous for  $B$ ;  $x, y \in A \cap B \Rightarrow x, y \in A \wedge x, y \in B \Rightarrow \overline{xy} \subset A \wedge \overline{xy} \subset B \Rightarrow \overline{xy} \subset A \cap B \Rightarrow A \cap B$  convex.

<sup>3</sup>The elements of  $\mathcal{R}$  are finite unions of the elements of  $\mathcal{K}$ . Finite unions of these polyconvex sets are still finite unions of elements of  $\mathcal{K}$ , i.e. polyconvex sets. Finite intersections of polyconvex sets can be transformed to finite unions of finite intersections of elements of  $\mathcal{K}$  using the distributive law of set algebra. These finite intersections are elements of  $\mathcal{K}$  and their finite union yields a polyconvex set.

the Minkowski functionals quantify complementary structure features and provide independent information.

The additivity of the Minkowski functionals provides a natural continuation of the functionals to the domain of polyconvex sets. By definition all polyconvex bodies can be constructed from unions of convex bodies and equation (2.1) provides an algorithm to evaluate such unions. By repeated application of the additivity, all finite unions can be transformed to sums of functional values of convex bodies and intersections of convex bodies. As  $\mathcal{K}$  is closed under intersections, the resulting formula may be evaluated over  $\mathcal{K}$  and provides a unique value for every body of  $\mathcal{R}$ . Using this continuation the Minkowski functionals can without loss of generality be treated like functions  $M_\nu : \mathcal{R} \rightarrow \mathbb{R}$ . The continued functionals are by construction additive over  $\mathcal{R}$  and it is easy to show that they are also motion-invariant over the enlarged domain. Equation (2.3), on the other hand, does not need to hold over all  $\mathcal{R}$ , it is required only over  $\mathcal{K}$  and hence it is referred to as *convex continuity* in the following.

Thus, Hadwiger's theorem can also be stated like this: Every additive, motion-invariant and convex continuous functional  $M : \mathcal{R} \rightarrow \mathbb{R}$  is a linear combination of Minkowski functionals.<sup>4</sup> This implies that all information of all possible mappings from polyconvex bodies to scalars — especially mappings quantifying the morphology of polyconvex bodies — is contained in the information given by  $d + 1$  Minkowski functionals, as long as the mappings are required to be additive, motion-invariant and convex continuous.

While the restriction to additive, motion-invariant and convex continuous functionals may seem rather arbitrary and limiting, the advantages usually outweigh the disadvantages. Additivity allows to build efficient algorithms, as large bodies can be treated like a union of smaller parts, which allows to break down the computation to local contributions. In addition to this computational advantage, additivity is equivalent to requiring that disjoint bodies can be evaluated independently and that the global structure value is just the sum of these disjoint contributions. This requirement does not seem quite as arbitrary and it seems reasonable to look for functionals which do not change completely when going from two disjoint individual objects to a

---

<sup>4</sup>The restriction of  $M$  to  $\mathcal{K}$  is a linear combination of Minkowski functionals in virtue of the previous formulation of Hadwiger's theorem and the additivity of  $M$  and the Minkowski functionals over  $\mathcal{R}$  assure that this linear combination yields the same values as  $M$  for all polyconvex bodies.

common evaluation of the disjoint objects. The required motion invariance causes the functionals to be independent of the chosen coordinate system, and convex continuity allows to approximate bodies by similar shapes, which is vital to keep binning and quantization errors acceptable.

### 2.1.2 Geometric Interpretation

While the basic properties of the Minkowski functionals have been introduced in the last section, it is not yet clear how to calculate or interpret them. Actually the interpretation can partially be guessed by using the homogeneity given in equation (2.5).  $M_0(\lambda A) = \lambda^d M_0(A)$ , i.e.  $M_0$  scales like a  $d$ -dimensional volume and actually is proportional to the volume of the body  $A$ . The volume is an additive, motion-invariant and continuous functional on the set of measurable sets, which is a superset of  $\mathcal{R}$ , and therefore the volume fits the characteristics of a Minkowski functional. Similarly,  $M_1$  scales like a surface and is proportional to the surface of the body. Another very interesting functional is  $M_d$ , which is scale invariant. The Euler characteristic  $\chi$  is a scale invariant, additive, motion-invariant and convex continuous functional and thus has to be proportional to  $M_d$ . As this topological constant is not quite as common as volumes and surfaces, a definition might be helpful (cf. Hadwiger, 1957):

$$\chi(A) = \begin{cases} 1, & A \neq \emptyset \wedge A \in \mathcal{K} \\ 0, & A = \emptyset. \end{cases} \quad (2.6)$$

For all other polyconvex bodies the value can be deduced via the additivity of  $\chi$ . So, the Euler characteristic counts how many disjoint convex bodies are present in a given configuration. If there are non-convex bodies in the configuration, the interpretation for  $d = 2$  is still quite instructive:

$$\chi(A) = \#_c(A) - \#_h(A), \quad (2.7)$$

where  $\#_c(A)$  counts the number of connected components and  $\#_h(A)$  the number of holes in these components. For  $d = 3$  there are different kinds of holes; they may be open like the center of a torus or completely covered internal cavities. Let  $\#_i(A)$  be the number of internal cavities and  $\#_o(A)$  the number of open holes, then

$$\chi(A) = \#_c(A) - \#_o(A) + \#_i(A). \quad (2.8)$$

While volume and surface are strictly non-negative, the codomain of the Euler characteristic comprises positive and negative values. This is a direct consequence of the negative term in equation (2.1), i.e. of the contribution of intersecting bodies to additive functionals.

A very general way to write down the Minkowski functionals is actually based exclusively on the Euler characteristic:

$$M_\nu(A) \propto \int \chi(A \cap E_\nu) \, d\mu(E_\nu), \quad (2.9)$$

where  $E_\nu$  denotes a  $\nu$ -dimensional affine subspace of  $\mathbb{R}^d$ , the integral runs over all motions (translations and rotations) of  $E_\nu$  and  $d\mu(E_\nu)$  is related to the Haar measure on the group of motions  $\mathcal{G}$  (Hadwiger, 1957). This abstract formulation is a generalization of the common principle to define a volume as integral over the characteristic function:

$$V(A) = \int_{\mathbb{R}^d} \mathbb{1}_A(\vec{x}) \, d^d x \quad (2.10)$$

$$\mathbb{1}_A(\vec{x}) = \begin{cases} 1, & \vec{x} \in A \\ 0, & \text{else} \end{cases} \quad (2.11)$$

Equation (2.10) is equivalent to equation (2.9) for  $\nu = 0$ , as  $E_0$  is a point, i.e. always convex, and  $A \cap E_0 = E_0$  if  $E_0 \subset A$ ,  $A \cap E_0 = \emptyset$  else. Therefore,  $\chi(A \cap E_0) = \mathbb{1}_A(\vec{x})$  with  $E_0 = \{\vec{x}\}$ . For larger  $\nu$  the object under study is not scanned by a testing point, but by testing lines, testing planes and testing hyperplanes. As these testing probes are extended, they need to be rotated in addition to be shifted over the object under study in order to obtain motion-invariant results. It is not obvious that probing an object with a line is proportional to the surface of the object, but looking more closely at equation (2.9) for  $\nu = 1$  reveals that the integral “counts” how often a line may intersect a given object. In order to intersect, the line has to enter the object somewhere at its boundary, and the surface of the object is just a measure for all possible entry points of a testing line. Thus, while equation (2.9) may not be too convenient in order to actually calculate Minkowski functionals, it provides a first geometric interpretation of these functionals (and is very useful from a theoretical point of view).

Another way to compute the Minkowski functionals is given by curvature



integrals (Santaló and Kac, 2004). For  $\nu \geq 1$  you find

$$M_\nu(A) \propto \int_{\partial A} S_{\nu-1}(k_1, \dots, k_{d-1}) \, d\mathcal{S}, \quad (2.12)$$

where  $\partial A$  denotes the boundary of  $A$ ,  $S_i$  the  $i^{\text{th}}$  elementary symmetric polynomial,<sup>5</sup>  $d\mathcal{S}$  the  $d-1$ -dimensional surface element and  $k_i$  the  $i^{\text{th}}$  principal curvature of  $\partial A$  at the current surface element. Obviously the given formula is only applicable if  $\partial A$  is smooth enough to actually compute its principal curvatures at every point of the surface, while equation (2.9) did not require smooth boundaries. This discrepancy can be solved by constructing a sequence of bodies  $A_n$  with sufficiently smooth boundary, which converges to  $A$  for large  $n$ . While the curvature integral can be evaluated for these  $A_n$ , the convex continuity of the Minkowski functionals can be used — if the sequence is chosen appropriately — to derive the functional values for  $A$ . How to construct such a sequence is e.g. shown in Mecke (1994, 1998).

The formulation via curvature integrals immediately yields a geometric interpretation for  $M_2$ , as it is proportional to the integrated mean curvature of a body. For  $d = 3$   $M_3$  correspondingly is proportional to the integrated Gaussian curvature. As higher dimensions are inherently not very depictive, this concludes the discussion of geometric analogs. Nevertheless, equations (2.9) and (2.12) are also valid for higher dimensions and should be helpful when trying to interpret other Minkowski functionals.

### 2.1.3 Summary

As the previous sections are quite mathematical and formal, here a short overview of the most important facts about Minkowski functionals:

- they are additive, motion-invariant and convex continuous
- there are  $d + 1$  of them in  $d$  dimensions
- they are complete, i.e. all other additive, motion-invariant and convex continuous functionals are linear combinations of Minkowski functionals and contain no further information

---

<sup>5</sup>The  $i^{\text{th}}$  elementary symmetric polynomial is a polynomial of order  $i$ , which consists of  $\binom{d-1}{i}$  different summands, where each summand is the product of  $i$  different arguments of the function.

- they scale like  $\lambda^{d-\nu}$
- they are proportional to an integral “counting” the possible intersections of an affine subspace with the given body
- they are proportional to curvature integrals
- in 2 dimensions they are proportional to area, perimeter and Euler characteristic
- in 3 dimensions they are proportional to volume, surface area, integrated mean curvature and Euler characteristic

### 2.1.4 Functionals for Pixel Images

There are several different ways to compute Minkowski functionals and two of them have been introduced in the previous sections. Whichever formulation you like most, due to their origin in integral geometry and close relation to measure theory, they all involve complicated integral expressions. A common technique to handle complex integrals is finding ways to avoid the integration. Working with Minkowski functionals this is actually quite often feasible. Suppose you want to characterize the shape of the black region in a 2-dimensional black and white (b/w) pixel image. You only have to compute the Minkowski functionals for a single pixel and the possible intersections of pixels, i.e. edge lines and corners points, and by virtue of the additivity you can compute Minkowski functionals of arbitrary 2-dimensional pixel images by just summing the right building blocks instead of solving integrals. So, whenever it is possible to reduce a given problem to basic building blocks, the additivity can help to prevent complicated integrals. If the chosen building blocks are very simple, like e.g. square pixels, the remaining integrals to evaluate the building blocks are usually a lot easier to solve than the initial problem.

Before discussing useful building blocks to characterize structures in gamma-ray counts maps, you need to find a way to apply the Minkowski functionals to the given data at all. A gamma-ray counts map is equivalent to a 2-dimensional gray scale image, but it is not a priori clear how to map such an image to the domain of polyconvex bodies. A common method to process gray scale data in statistical physics (e.g. Mantz et al., 2008) is simple thresholding,

i.e. choosing a threshold  $\rho$  in the range of gray levels and setting all pixels in the image to white if their gray level is below the threshold, and to black if their gray level is at least as big as the chosen threshold. After thresholding the black parts of the image are the polyconvex body  $K(\rho)$  which may be studied using Minkowski functionals. As the shape of  $K(\rho)$  depends on the chosen threshold  $\rho$ , you need to study the morphology at every threshold to fully characterize the structure of the gray scale image.

So, using thresholding the afore mentioned square pixels of b/w pixel images are great building blocks of an efficient morphometric analysis of gamma-ray counts maps.<sup>6</sup> In a 2-dimensional square lattice, every pixel has 8 next neighbors (4 at the pixel edges and 4 at the pixel corners). As black regions are supposed to be connected if neighboring pixels are black, they need to touch, i.e. black pixels intersect neighboring black pixels, resulting in edge lines or corner points as intersection bodies. As these are the only possible intersections on a regular square lattice, it is sufficient to determine the Minkowski functionals of pixels, edge lines and corner points to be able to process the entire pixel image based on the additivity.

As the Minkowski functionals in  $\mathbb{R}^2$  are proportional to area  $A$ , perimeter  $P$  and Euler characteristic  $\chi$ , these functional are used in the remaining sections instead of referring to  $M_0$ ,  $M_1$  and  $M_2$ . The relevant values of these functionals for pixels, edge lines and corner points are given in table 2.1. While most of these values should be self explanatory, it might be confusing to assign the perimeter length 2 to an edge line of length 1. This value is needed to satisfy additivity<sup>7</sup> and convex continuity,<sup>8</sup> but it is actually not very artificial as it corresponds to the length of the shortest path around the

---

<sup>6</sup>There are other possible approaches, like the marching square algorithm (Mantz et al., 2008). These algorithms usually reduce binning effects, e.g. by adding diagonal edges to the square lattice. As the motivation of this thesis is to characterize typical background structures and to quantify deviations and *not* to extract source morphologies from the data via Minkowski functionals, it is not quite as important to reduce binning effects. Respectively, it is not vital for a binned circle to converge to the Minkowski functionals of a circle; as long as the given counts map is compared to the expectations of a binned background measurement, binning effects are a secondary concern.

<sup>7</sup>Let  $K_1$  and  $K_2$  be two pixels touching at an edge.  $P(K_1 \cup K_2) = 6$  and  $P(K_1) = P(K_2) = 4$ , so  $P(K_1 \cap K_2) = 2$  is needed to fulfill additivity.

<sup>8</sup>Let  $K_n$  be a rectangle of length 1 and width  $1/n$ .  $K_n$  is convex and it converges to an edge line for large  $n$ , so the perimeter of an edge line needs to be the same as  $\lim_{n \rightarrow \infty} P(K_n) = \lim_{n \rightarrow \infty} 2 + 2/n = 2$ .

	$A$	$P$	$\chi$
pixel	1	4	1
edge line	0	2	1
corner point	0	0	1

Table 2.1: Values of area  $A$ , perimeter  $P$  and Euler characteristic  $\chi$  for pixels, edges lines and corners points in units of the lattice constant.

given line.

Knowing the basic building blocks the algorithm to evaluate a b/w image is straightforward. You need 2 lists for book keeping. A list of unprocessed pixels, initially containing all the pixels of the image, and a list of processed pixels, initially empty. Take a pixel from the list of unprocessed pixels; if the pixel is white, put it in the list of processed pixels and continue with the next pixel; if the pixel is black, add the pixel contributions from table 2.1 to the current functional values and check if one of its neighboring pixels is already in the list of processed pixels and black; if there is such a neighboring pixel, remove the contribution of the intersection from the current functional values; put the pixel in the list of processed pixels and continue with the next pixel.

The bookkeeping can be greatly simplified by fixing the order in which the pixels are processed. All pixels before the current pixel are already processed and all following pixels unprocessed. By using a scheme like depicted in figure 2.1, you can even be sure that only the 4 pixels below or left to the current pixel may ever contribute relevant intersections, as only these pixels are already part of your processed polyconvex body. Actually, all relevant contributions can even be determined by using the highlighted  $2 \times 2$  region of figure 2.1. While this region neglects possible intersections with the pixel to the lower right of the current pixel, these contributions can be corrected for in the next  $2 \times 2$  region. So if the current pixel is black (otherwise there are no intersections) and the pixel below the current pixel is white (otherwise the lower right corner of the current pixel is already part of an intersection and thus accounted for), the highlighted  $2 \times 2$  region might erroneously neglect intersections with a black pixel to the lower right. This error prone configuration can easily be identified in the next  $2 \times 2$  region: The error occurred if and only if the upper left pixel is black, the lower left pixel is white

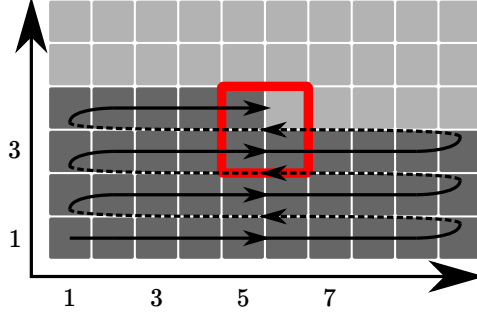










Figure 2.1: Schematic view of the pixel coordinate system and the order in which the contributions of individual pixels to the overall functional values are determined. The pixels are just numbered, starting in the lower left corner. Pixel contributions are evaluated line by line from lower left to upper right. Dark gray pixels are already processed, while light gray pixels depict unprocessed pixels. The red frame highlights the  $2 \times 2$  pixels region, which can be used to determine the contributions of the current pixel.

and the lower right pixel is black. Thus, when finding such a configuration the contribution of a corner edge point has to be subtracted in addition to the current pixels contribution, to correct the contribution of the previous pixel.

This allows to precompute all possible contributions of a pixel by evaluating the  $2^4$  different configurations of a  $2 \times 2$  pixel region (where the upper right pixel corresponds to the current pixel). Thus, instead of adding single pixels and looking for intersections in every step of the algorithm, you may slide a  $2 \times 2$  pixel window across the image and just look up the contribution of the upper right pixel in a table of the 16 possible configurations of this window. The values needed to implement such a lookup table are given in table 2.2.

While this in principle solves the problem to determine the contributions of single pixels, it is not the most convenient mapping, as it assigns very different values to symmetric configurations, which does not reflect the inherent motion invariance of the Minkowski functionals. In order to derive a more appealing mapping, you have to resort to the interpretation of Minkowski functionals as curvature integrals. According to this point of view, the Euler characteristic is proportional to the integrated curvature, which in turn is proportional to the

	$A$	$P$	$\chi$
	0	0	0
	0	0	0
	0	0	0
	0	0	0
	0	0	0
	0	0	0
	0	0	-1
	0	0	0









	$A$	$P$	$\chi$
	1	4	1
	1	4	0
	1	2	0
	1	2	0
	1	2	0
	1	2	0
	1	0	-1
	1	0	0









Table 2.2: Possible contributions of a pixel to the area  $A$ , the perimeter  $P$  and the Euler characteristic  $\chi$ , based on its lower left neighbors (cf. figure 2.1). Pixels which are part of the polyconvex body are highlighted brown, exterior pixels are depicted gray. Corrections to the contribution of the previous pixel are included (see text for details).

number of closed curves. Thus, by assigning every black corner a contribution of  $1/4$  and every white corner, i.e. dip in a black body, a contribution of  $-1/4$ , the Euler characteristic can be computed as a sum of local curvature contributions (Mecke, 1994). By simultaneously counting contributions to the perimeter, i.e. edges between black and white pixels, directly instead of inferring it from intersection counts and pixel contributions and counting the number of black pixels, the contributions of a  $2 \times 2$  region take the form given in table 2.3. Even though table 2.2 and 2.3 look very different, they yield the same functional values when applied to the entire polyconvex body. Due to the higher symmetry of table 2.3 and the increased flexibility,<sup>9</sup> the scheme based on local curvatures is used in the remaining discussion.

While this solves the main problems of how to determine Minkowski functionals of gamma-ray counts maps, there are still some essential subtleties left. As discussed in the previous paragraphs, the next neighbors of a pixel are essential to determine the correct contribution of a pixel to the different structure functionals. What is missing so far, is a way to compute the contri-

---

<sup>9</sup>You do not need to process the pixels from lower left to upper right, you just need to add all contributions of  $2 \times 2$  pixel regions.

	$A$	$P$	$\chi$
	0	0	0
	1/4	1	1/4
	1/4	1	1/4
	1/2	1	0
	1/4	1	1/4
	1/2	1	0
	1/2	2	-1/2
	3/4	1	-1/4









	$A$	$P$	$\chi$
	1/4	1	1/4
	1/2	2	-1/2
	1/2	1	0
	3/4	1	-1/4
	1/2	1	0
	3/4	1	-1/4
	3/4	1	-1/4
	1	0	0

Table 2.3: Possible contributions of a  $2 \times 2$  pixel region to the area  $A$ , the perimeter  $P$  and the Euler characteristic  $\chi$ , based on the local perimeter and curvature contributions. Pixels which are part of the polyconvex body are highlighted brown, exterior pixels are depicted gray. As the contribution of every pixel to  $A$  is counted 4 times (in 4 different overlapping  $2 \times 2$  regions), the contributions of single pixels are divided by 4; similarly every edge is counted twice and thus edge contributions are divided by 2. Curvature contributions are localized at the center of the  $2 \times 2$  and are not weighted, as they are counted only once.

bution if some of these neighbors are missing, i.e. a discussion of boundary conditions. If a black region touches the boundary of the image, it is not clear if these edges are part of the regions perimeter or if the region extends beyond the visible image and the boundary edges are internal points of this larger body. As it is in general not possible to predict the surrounding of a given image, you'll have to apply some boundary conditions to fix the contribution of boundary pixels. There are many possible choices and two common and representative examples of possible boundary conditions are discussed in the following paragraphs.

### Minus Sampling Boundary Conditions

















Minus sampling boundary conditions sacrifice a part of the given image to fix the contribution of boundary pixels. By using only the rows and columns 2 to  $N - 1$  of an  $N \times N$  pixel image, the outer most pixels can be used to determine

the contribution of the boundary pixels of the inner  $(N - 2) \times (N - 2)$  region. This way the contributions of the different boundary configurations can easily be determined; in fact they are very similar to the ones given in table 2.3, the only difference being that all areas and edges outside the inner image region are ignored, i.e. their contribution is set to 0. The resulting contributions for minus sampling boundary conditions are given in table 2.4.

As the current mapping computes the Euler characteristic via local curvatures, i.e. counts the number of closed perimeters, there may be fractional values for  $\chi$  in the final result, i.e. only partly closed perimeters. A convex body being cut in two pieces by one of the boundary lines contributes a value of  $1/2$  to  $\chi$  and a convex body which is cut at the corner of the image may even contribute  $1/4$  to  $\chi$ . This just corresponds to the decision to account for partial bodies and to allow them to extend beyond the image. If this is not acceptable, you need to choose different boundary conditions, like setting all pixels on the outside of the current image to white. This prevents partial bodies by forcefully closing all of them at the boundaries.

While the mapping in table 2.4 is kind of natural and simple, it is not the only possible choice. Suppose the given image is just a clipped view of a larger area and you want to be able to add up the contributions of several neighboring images to obtain the Minkowski functionals of the entire area. Given the current mapping, you need to reprocess all boundaries, as every image fully accounts for perimeter sections and curvatures at the boundaries, i.e. you need to subtract doubly counted boundary structures when combining neighboring views. If such unions of separately processed images are a common task, it may be beneficial to choose a different mapping for the boundary contributions, where perimeter sections and curvatures at the boundary are divided equally among all neighboring images and only a fraction of the entire contribution is added to each image. This way the Minkowski functionals of the different images can be added without reprocessing the boundaries and the sum immediately yields the structure values of the larger region. The disadvantage of such a convention are the fractional contributions of boundary sections to the Minkowski functionals of the different views. While these are irrelevant for some applications, they might lead to inconsistencies



	$A$	$P$	$\chi$
	0	0	0
	0	0	1/4
	0	1/2	1/4
	0	1/2	0
	0	1/2	1/4
	0	1/2	0
	0	1	-1/2
	0	1	-1/4
	1/4	1	1/4
	1/4	1	-1/2
	1/4	1/2	0
	1/4	1/2	-1/4
	1/4	1/2	0
	1/4	1/2	-1/4
	1/4	0	-1/4
	1/4	0	0

















	$A$	$P$	$\chi$
	0	0	0
	0	1/2	1/4
	0	1/2	1/4
	0	1	0
	1/4	1	1/4
	1/4	1/2	0
	1/4	3/2	-1/2
	1/4	1	-1/4
	1/4	1	1/4
	1/4	3/2	-1/2
	1/4	1/2	0
	1/4	1	-1/4
	1/2	1	0
	1/2	1/2	-1/4
	1/2	1/2	-1/4
	1/2	0	0

Table 2.4: Possible contributions of a  $2 \times 2$  pixel region at the boundaries of the image to the area  $A$ , the perimeter  $P$  and the Euler characteristic  $\chi$ , based on the local perimeter and curvature contributions. Pixels which are part of the polyconvex body are highlighted brown, exterior pixels are depicted gray. Hatched pixels correspond to the outer most line of pixels and are used to determine the boundary contributions of interior pixels via minus sampling boundary conditions. They are treated like being outside of the current image and are used only to provide an embedding environment in order to clarify perimeter and curvature contributions at the boundaries. The fractional values account for repeated evaluations of the same contributions (cf. figure 2.3 for details). The contributions of the rotated regions, i.e. boundary regions on other sides of the image, are the same and not listed separately.

for others.<sup>10</sup>

The contributions shown in table 2.4 are actually not only applicable to minus sampling boundary conditions, but to all boundary conditions which provide information about an additional row of pixels surrounding the given image. This includes setting the outside of the image to black, white or a check pattern and less static choices like reflected boundary conditions, i.e. filling each outer pixel with the same color as its inner neighbor.

### Periodic Boundary Conditions

While periodic boundary conditions could be implemented just like in the previous paragraphs, i.e. by filling the pixels outside of the image with repeated copies of the image, it is more common to implement them via a change of topology. Instead of filling outer pixels with a copy of the corresponding pixel on the other side of the image, the outer pixel is usually identified with this pixel, which effectively changes the topology of the image from a plane to a torus. The disadvantage of repeated copies is the fact that the corresponding boundaries (left/right and top/bottom) count the same perimeter sections and associated curvatures at the boundary, i.e. the boundary is counted twice. This may be solved by weighting the boundary contributions appropriately and processing all boundaries, or by the previously mentioned change of topology, which identifies corresponding boundaries and thus reduces the needed work by a factor of 2.

Figure 2.2 shows the relevant pixels to process when using periodic boundary conditions. The contributions of the different  $2 \times 2$  regions of the shown domain are simply given by table 2.3, i.e. you can just apply the contributions of internal pixels everywhere, including the boundary regions. This corresponds to the previously mentioned fact that periodic boundary conditions are equivalent to a transition from the flat image to the surface of a torus. As the torus is a manifold without boundary, there are no special pixels, which call for different contributions. All pixels are equal and have 8 unique neighbors among the other pixels of the image, so all contributions can be determined without further conditions.

Thus, periodic boundary conditions involve only two glue lines compared to

---

<sup>10</sup>If all but boundary contributions are integer valued, the fractional part of the final result contains special information about the boundary regions. Such a special tag for the state of the current boundary may cause inconsistencies for some applications.

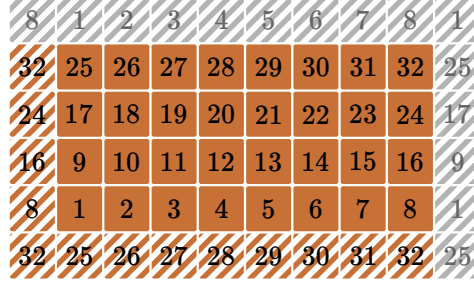


Figure 2.2: Schematic view of periodic boundary conditions. The solid brown pixels depict the original image region and a unique pixel ID is shown on each of them. The IDs shown on the hatched pixels are references to the image pixel which has to be cloned to the position of the hatched pixel in order to impose periodic boundary conditions. The brown region (solid and hatched pixels) shows the domain which needs to be processed in order to evaluate the Minkowski functionals including all contributions of periodic boundary conditions.

four boundary lines with minus sampling, they do not have to sacrifice image data to evaluate these glue lines and they seem to introduce less artificial structures than setting the exterior of the image to white or similar boundary conditions. Nevertheless, you have to be careful when using periodic boundary conditions, as the change of topology may severely influence the structure of the image. Consider e.g. the image of a smooth gradient. Periodic boundary conditions introduce a discontinuous jump at the glue lines and thereby introduce a pronounced artificial structure.

This concludes the short introduction into the wide and interesting field of Minkowski functionals. By now you should have a basic understanding of why Minkowski functionals are powerful tools to quantify structure and how to interpret them geometrically. In the last section it was shown how to apply them to gray scale images and how to incorporate different boundary conditions. While this already provides a solid base for studying the structure of gamma-ray counts maps, the following sections first discuss further useful tools like the generalization to Minkowski tensors and techniques to calculate probability distributions of Minkowski functionals. When these discussions

completed the toolbox of morphometric counts map analysis, applications and results are shown in chapter 6.

### 2.2 Minkowski Tensors and Alesker's theorem

While the Minkowski functionals provide a quite impressive framework to study structures in physics, you might wonder if their defining properties are too restrictive and can be relaxed.

You probably do not want to sacrifice the ability to approximate convex bodies with similar objects and therefore convex continuity should not be dropped and not even be relaxed (it is already only required for convex bodies and optional for other polyconvex objects).

As additivity is very essential for the underlying theory, i.e. for proving theorems about Minkowski functionals, and is actually a quite loose constraint compared to the  $\sigma$ -additivity required in measure theory, you probably should not drop it either (especially as it is very useful for computing Minkowski functionals and helps to avoid integrals, as seen in the last section).

The motion invariance guarantees coordinate system independent results, but it also causes the Minkowski functionals to be insensitive to orientation and position of bodies. The configurations shown in figure 2.3 are indistinguishable using Minkowski functionals. This weakness makes motion invariance the prime candidate to look for alternatives. However, as the result of structure quantification should not be dominated by the chosen coordinate system, it is reasonable to make sure that the influence of a change of the coordinate system is predictable, i.e. when changing the coordinate system you should be able to transform your results to the new system without needing to reevaluate them. If you change the required motion invariance to motion covariance, the structure values transform just like the coordinates under a change of the coordinate system and it could indeed be shown recently that the resulting class of tensor valued functionals is also spanned by just a few Minkowski tensors (McMullen, 1997; Alesker, 1999; Hug et al., 2008).

#### 2.2.1 General Properties

In order to understand the meaning of motion covariance in this context, it is useful to recapitulate the transformation rules of tensors under rotations and reflections. Let be  $\mathbf{R} \in O(d)$ , where  $O(d)$  is the orthogonal group of degree

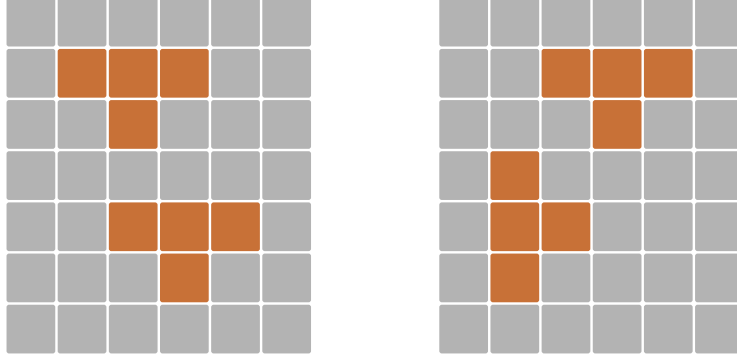


Figure 2.3: Two obviously different configurations, which are indistinguishable from the point of view of Minkowski functionals. Due to their motion invariance the Minkowski functionals are not sensitive to the position and orientation of bodies.

$d$ , i.e. the group of rotations and reflections in  $d$  dimensions. Let further  $\hat{R}$  be the associated rotation matrix and  $R_{ij}$  the components of this matrix.<sup>11</sup> Then the rotation and reflection of a vector  $\vec{x}$  is given by:

$$\begin{aligned}\vec{x}' &= \mathbf{R}\vec{x} \\ x'_i &= \sum_j R_{ij}x_j\end{aligned}\tag{2.13}$$

Similarly the rotation and reflection of a rank 3 tensor  $W$  with components  $W_{jik}$  is given by

$$\begin{aligned}W' &= \mathbf{R}W \\ W'_{ijk} &= \sum_{lmn} R_{il}R_{jm}R_{kn}W_{lmn},\end{aligned}\tag{2.14}$$

i.e. by transforming each tensor index individually. For general rank  $r$  tensors you have to transform each of the  $r$  indices individually. Let  $\mathcal{T}$  be the space of

---

<sup>11</sup>As the intended field of application are pixel images, i.e. flat euclidean spaces, there is a trivial relation between co- and contravariant tensor indices. Thus, co- and contravariant is used synonymous in this work and the position of a given index is a mere choice of space rather than a statement of transformation behavior. For details about Minkowski tensors on more general manifolds see Alesker (2007, and references therein).

$d$ -dimensional tensors<sup>12</sup> with components in  $\mathbb{R}$ , and  $\mathcal{R}$  the set of  $d$ -dimensional polyconvex bodies. A functional  $W : \mathcal{R} \rightarrow \mathcal{T}$  is called rotation-covariant if

$$W(\mathbf{R}A) = \mathbf{R}W(A), \quad (2.15)$$

for all orthogonal transformations  $\mathbf{R} \in O(d)$  and polyconvex bodies  $A \in \mathcal{R}$ .

Before going into details about translation-covariance, it is useful to have a quick look at the canonical building blocks of higher rank tensors. By using the tensor product  $\otimes$ , it is possible to build higher rank tensors from lower rank tensors. Instead of a formal definition of  $\otimes$  the following example should be sufficient to understand its basic properties. Let  $X$  be a rank 2 tensor and  $Y$  a rank 3 tensor, then the components of the rank 5 tensor  $W = X \otimes Y$  are given by

$$W_{ijklm} = X_{ij}Y_{klm}, \quad (2.16)$$

i.e. by multiplying corresponding components of  $X$  and  $Y$ . As  $\otimes$  is not commutative ( $X_{ij}Y_{klm} = Y_{klm}X_{ij}$  which is in general different from  $Y_{ijk}X_{lm}$ ), it is often handy to use the symmetric tensor product  $\odot$ . It is defined by

$$X \odot Y := \frac{1}{2}(X \otimes Y + Y \otimes X). \quad (2.17)$$

Using these operations it is possible to e.g. build a rank 2 tensor from two rank 1 tensors (i.e. vectors)  $\vec{a}$  and  $\vec{b}$  via  $\vec{a} \otimes \vec{b}$  or  $\vec{a} \odot \vec{b}$ . Besides vectors, common building blocks for higher rank tensors include the metric tensor and the unit tensors.<sup>13</sup> As it is quite common that some of these building blocks appear several times in the same tensor product, it is convenient to introduce the following notation:

$$X^{\odot n} := \underbrace{X \odot X \odot \cdots \odot X}_{n \text{ times}} \quad (2.18)$$

Now consider the rank  $r + k$  tensor  $\vec{a}^{\odot r} \odot \vec{x}^{\odot k}$ , where  $\vec{a}$  is some translation-invariant vector (e.g. a vector of a homogeneous force field) and  $\vec{x}$  is the

---

<sup>12</sup> $\mathcal{T} = \oplus_{r=0}^{\infty} \mathcal{T}^r$ , where  $\mathcal{T}^r$  denotes the space of  $d$ -dimensional rank  $r$  tensors, i.e.  $\underbrace{d \times d \times \cdots \times d}_{r \text{ times}}$  components.

<sup>13</sup>Unit tensors are tensors where all components are 0 except for components where all indices are equal, these are 1.

position of a point. Displacing the point by a translation  $\vec{t}$  changes the tensor like

$$\begin{aligned}\vec{a}^{\odot r} \odot (\vec{x} + \vec{t})^{\odot k} &= \vec{a}^{\odot r} \odot \left( \sum_{i=0}^k \binom{k}{i} \vec{x}^{\odot k-i} \odot \vec{t}^{\odot i} \right) \\ &= \sum_{i=0}^k \binom{k}{i} \vec{a}^{\odot r} \odot \vec{x}^{\odot k-i} \odot \vec{t}^{\odot i},\end{aligned}\tag{2.19}$$

i.e. the result is a linear combination of lower rank tensors multiplied with powers of  $\vec{t}$ . As the given example is about the simplest combination of basic building blocks, it represents the typical form of the simplest translational transformation you can get. Thus, translation-covariance of a functional  $W : \mathcal{R} \rightarrow \mathcal{T}$  is given, if

$$W(A + \vec{t}) = \sum_{i=0}^k W_i(A) \odot \vec{t}^{\odot i},\tag{2.20}$$

where  $0 \leq k < \infty$  and  $W_i : \mathcal{R} \rightarrow \mathcal{T}$  are functions associated with  $W$ . If a functional  $W : \mathcal{R} \rightarrow \mathcal{T}$  is translation- and rotation-covariant, it is called motion-covariant.

From the previous examples it should be obvious that  $\vec{x}^{\odot r}$  is a motion-covariant tensor of rank  $r$ . This building block can be used to generalize the volume  $V(A) = \int_A d^d x$  to the moment tensors

$$V^r(A) = \int_A \vec{x}^{\odot r} d^d x.\tag{2.21}$$

$V^r$  is a tensor of rank  $r$ , which is additive, convex continuous and motion-covariant. Assuming a homogeneous density of 1 for the body  $A$  the moment tensors furthermore describe important physical properties. The center of mass is e.g. given by  $\vec{x}_{\text{cm}}(A) = V^1(A)/V^0(A)$  and the moment of inertia tensor is given by  $I(A) = \text{tr}(V^2(A))\mathbb{1}_d - V^2(A)$ , where  $\mathbb{1}_d$  is the rank 2 unit tensor in  $d$  dimensions and  $\text{tr}(V^2(A))$  the trace of the second moment tensor. While  $V^1$  contains information about the position of the object, the central moments

$$\begin{aligned}V_c^r(A) &= \int_A (\vec{x} - V^1(A))^{\odot r} d^d x \\ &= \sum_{i=0}^r (-1)^i \binom{r}{i} V^{r-i}(A) \odot (V^1(A))^{\odot i}\end{aligned}\tag{2.22}$$

contain information about orientation and shape of the object. In fact, given the probability density  $\rho(\vec{x}) = \mathbb{1}_A(\vec{x})/V^0(A)$ , where  $\mathbb{1}_A$  is the characteristic function of  $A$  (cf. equation (2.11)), the central moments of this uniform distribution are given by  $V_c^r(A)/V^0(A)$  and thus the tensors  $V_c^r(A)$  describe the shape of  $A$  in the same way as central moments of a probability distribution describe this distribution.<sup>14</sup> So, the eigenvalues of  $V_c^2(A)$  determine the sizes of the major axes of an ellipsoid, which is similar to  $A$  in size and orientation, and  $V_c^3(A)$  contains information about the skewness, i.e. asymmetry, of  $A$  relative to its center of mass.

Even in this first set of tensors, based exclusively on a single Minkowski functional, there is a wealth of information.<sup>14</sup> Nevertheless, it is actually possible to construct similar families of tensors for each Minkowski functional. The easiest way to see this, is by using the formalism of curvature integrals from equation (2.12). The only difference to the tensors  $V^r$  is the existence of a second relevant vector besides the position vector  $\vec{x}$ . As all the integrals in equation (2.12) are surface integrals, they may additionally include normal vectors of the surface  $\vec{n}$ . So in general you can construct the following Minkowski tensors:

$$M_\nu^{p,q}(A) \propto \int_{\partial A} S_{\nu-1}(k_1, \dots, k_{d-1}) \vec{x}^{\odot p} \odot \vec{n}^{\odot q} d\mathcal{S} \quad (2.23a)$$

for  $\nu \geq 1$  and — for the sake of a uniform notation —

$$M_0^{p,q}(A) \propto \begin{cases} V^p(A), & \text{if } q = 0 \\ 0, & \text{else.} \end{cases} \quad (2.23b)$$

These tensors are homogeneous functionals under scaling of  $A$ . Their scaling behavior is determined by the underlying Minkowski functional and the power of  $\vec{x}$ . Therefore,

$$M_\nu^{p,q}(\lambda A) = \lambda^{d-\nu+p} M_\nu^{p,q}(A). \quad (2.24)$$

Each of the families of Minkowski tensors, i.e. the tensors of different  $\nu$ , describes the body  $A$  quite in detail,<sup>14</sup> but each of them from a different

---

<sup>14</sup>If the characteristic function of a probability distribution exists and is holomorphic at the origin, its series expansion is fully determined by the moments of the probability distribution (Billingsley, 1995). So, knowing all moments of a probability distribution allows (under certain conditions) to construct its characteristic function, which allows to fully reconstruct the initial distribution.



point of view. E.g while  $M_0^{1,0}(A)$  gives the center of mass of the entire body,  $M_1^{1,0}(A)$  gives the center of mass of its surface and tensor families for larger  $\nu$  describe the distributions of different curvatures. Even when McMullen (1997) first introduced these tensors and formulated the conjecture that they might span the space of motion-covariant valuations, he already noted that the tensors are linearly dependent. So, the information of different families is not completely complementary, but assuming different points of view can help to increase the information output without needing to resort to higher rank tensors.

Alesker (1999) was able to prove the completeness of the Minkowski tensors. He could show that every additive, convex continuous and motion-covariant functional  $W : \mathcal{K} \rightarrow \mathcal{T}$ , which contains only a finite number of different tensor ranks in its image,<sup>15</sup> can be written as a linear combination of tensors of the form  $\mathbb{1}_d^{\odot s} \odot M_\nu^{p,q}(A)$ , where  $\mathbb{1}_d$  is the rank 2 unit tensor in  $d$  dimensions and  $M_\nu^{p,q}(A)$  are the Minkowski tensors from equation (2.23). Just like with the Minkowski functionals the additivity can be used to promote this remarkable result from  $\mathcal{K}$  to  $\mathcal{R}$ , i.e. to the set of polyconvex bodies. Thus, the Minkowski tensors contain all the information of polyconvex bodies you could possibly quantify using additive, convex continuous and motion-covariant functionals, as all of them can be reduced to a linear combination of Minkowski tensors.

While the definition of motion covariance is quite general and covers a large set of valuations, the transformation properties of Minkowski tensors are simpler and can readily be written down:

$$\begin{aligned} M_\nu^{p,q}(\mathbf{R}A) &= \mathbf{R}M_\nu^{p,q}(A) \\ M_\nu^{p,q}(A + \vec{t}) &= \sum_{i=0}^p \binom{p}{i} M_\nu^{p-i,q}(A) \odot \vec{t}^{\odot i}, \end{aligned} \quad (2.25)$$

where  $\mathbf{R} \in O(d)$  is an orthogonal transformation and  $\vec{t}$  a translation vector. As these formulas contain only tensors whose  $p$ -index is less than or equal to the  $p$ -index of the initial tensor, stating the results of a structure analysis by giving all Minkowski tensors up to a certain  $p$ -index in addition to the chosen

---

<sup>15</sup>Usually the most interesting functionals contain exactly 1 tensor rank in their image, i.e. usually you are interested in mappings to tensors of a certain rank. Nevertheless, Alesker's theorem is even valid for mappings which return tensors of different ranks for different bodies or mappings that return sums of tensors of different ranks, as long as there are finitely many different ranks involved.

coordinate system allows to transform the results to any other coordinate system<sup>16</sup> immediately and in this sense the results are universal.

As mentioned before, the Minkowski tensors are not entirely independent, but Hug et al. (2008) have shown that there are just a limited number of linear relations between the tensors. They also demonstrate how to construct a set of independent tensors and give examples of independent sets of tensors up to rank 3 (see Hug et al. (2008) for details).

This concludes the general introduction of the Minkowski tensors. As the Minkowski functional are the Minkowski tensors of rank 0, i.e.  $M_\nu = M_\nu^{0,0}$ , the Minkowski functionals are no longer treated separately and the following discussion focuses exclusively on Minkowski tensors.

### 2.2.2 Summary

As the previous sections are quite mathematical and formal, here a short overview of the most important facts about Minkowski tensors:

- they are additive, motion-covariant and convex continuous
- they describe higher moments of the underlying Minkowski functionals
- they are complete, i.e. all other additive, motion-covariant and convex continuous functionals are linear combinations of Minkowski tensors and contain no further information
- they scale like  $\lambda^{d-\nu+p}$
- there are only a few of them, if you restrict your analysis to tensors of a certain rank
- their local contributions depend on position and normal vectors, i.e. on the coordinates and orientations of surface sections

### 2.2.3 Tensors for Pixel Images

Computing Minkowski tensors for 2-dimensional b/w pixel images is in fact very similar to computing Minkowski functionals of these images. Both are

---

<sup>16</sup>Here coordinate system means an arbitrarily chosen origin and an orthonormal basis, as skewed and stretched basis vectors change the inherent structure and are therefore unsuitable for structure analysis.

additive and so both can be reduced to sums over individual pixels. The only difference is the coordinate dependence of these contributions. For the following discussions the used coordinate system is a simple numbering of pixels from lower left to upper right, as shown in figure 2.1. The pixels are centered at the integer valued lattice points and the units of the axes are chosen such that the length of each square pixel is 1. This leads to the following contribution of a pixel at  $(x, y)$  to the area tensor  $A^p \propto M_0^{p,0}$ :

$$\begin{aligned} A_{i_1 i_2 \dots i_p}(x, y) &= \int_{x-0.5}^{x+0.5} \int_{y-0.5}^{y+0.5} x_1^k x_2^l dx_2 dx_1 \\ &= \frac{(x+0.5)^{k+1} - (x-0.5)^{k+1}}{k+1} \\ &\quad \frac{(y+0.5)^{l+1} - (y-0.5)^{l+1}}{l+1}, \end{aligned} \quad (2.26)$$

where  $A_{i_1 i_2 \dots i_p}$  are the components of  $A^p$  and the exponents  $k$  and  $l$  are counts of how often one of the indices  $i_m$  is 1 or 2 respectively, i.e.  $k = \sum_{m=1}^p \delta_{1i_m}$ ,  $l = \sum_{m=1}^p \delta_{2i_m}$ , where  $\delta_{ij}$  is the Kronecker delta.

Perimeter sections can be parametrized via  $\vec{x} = 1/2(\vec{a} + \vec{b}) + \lambda(\vec{b} - \vec{a})$ , where  $\vec{a}$  and  $\vec{b}$  are the start and end points of the perimeter section and  $\lambda \in [-1/2; 1/2]$ . As there are only horizontal and vertical perimeter sections, this can be further simplified to

$$\begin{aligned} x_1 &= x + \lambda \\ x_2 &= y \end{aligned} \quad (2.27a)$$

for horizontal perimeter sections and

$$\begin{aligned} x_1 &= x \\ x_2 &= y + \lambda \end{aligned} \quad (2.27b)$$

for vertical perimeter sections, where  $(x, y)$  are the coordinates of the center of the given section. The surface normal  $\vec{n}$  of such a perimeter section is constant and thus the contributions of perimeter sections to  $P^{p,q} \propto M_1^{p,q}$  are

given by

$$\begin{aligned}
 P_{i_1 i_2 \dots i_p}^{j_1 j_2 \dots j_q}(x, y) &= n_1^m n_2^n \int_{-0.5}^{0.5} x_1^k(\lambda) x_2^l(\lambda) d\lambda \\
 &= \begin{cases} n_1^m n_2^n y^l \frac{(x+0.5)^{k+1} - (x-0.5)^{k+1}}{k+1} \\ n_1^m n_2^n x^k \frac{(y+0.5)^{l+1} - (y-0.5)^{l+1}}{l+1} \end{cases} \quad (2.28)
 \end{aligned}$$











for horizontal (upper case) and vertical (lower case) perimeter sections respectively, where  $k$  and  $l$  are the counts of the different possible values of lower indices and  $m$  and  $n$  are the counts of the different possible values of upper indices, i.e.  $k = \sum_{\alpha=1}^p \delta_{1i_\alpha}$ ,  $l = \sum_{\alpha=1}^p \delta_{2i_\alpha}$ ,  $m = \sum_{\alpha=1}^q \delta_{1j_\alpha}$  and  $n = \sum_{\alpha=1}^q \delta_{2j_\alpha}$ .

















Just like with  $\chi$ , the contributions to  $\chi^{p,q} \propto M_2^{p,q}$  are localized at the corners of pixels and the local contribution of a corner is given by

$$\chi_{i_1 i_2 \dots i_p}^{j_1 j_2 \dots j_q}(x, y) = \chi n_1^m n_2^n x^k y^l, \quad (2.29)$$

where  $\chi$  is the usual local contribution to the Euler characteristic,  $\vec{n}$  is the sum of the surface normals of the incident perimeter sections,  $(x, y)$  the position of the corner and  $k, l, m$  and  $n$  the index counts analogous to the ones used for  $P^{p,q}$ .

Using the provided formulas you can immediately compute local contributions by just plugging in the proper coordinates, components of the normal vector and local curvatures. These details are given for each of the 16 possible configurations of a  $2 \times 2$  pixel region in table 2.5 for the area tensors  $A^p$ , in table 2.6 for the perimeter tensors  $P^{p,q}$  and in table 2.7 for the curvature tensors  $\chi^{p,q}$ . By summing the contributions of all  $2 \times 2$  regions in a given b/w pixel image you immediately get the full information of arbitrary sets of Minkowski tensors of the entire image.

	$\Delta x$	$\Delta y$
	0	0
	1	0
	0	0
	1	0
	0	1
	0	0
	0	1
	1	0
	0	1

	$\Delta x$	$\Delta y$
	0	0
	1	0
	0	1
	1	1
	0	0
	0	0
	1	1
	1	0
	1	1
	0	0
	0	0
	1	0
	1	1
	0	1
	1	1




















	$\Delta x$	$\Delta y$
	0	0
	0	1
	1	1
	1	0
	0	1
	1	1
	0	0
	1	0
	0	1
	1	1

Table 2.5: Contributions of a  $2 \times 2$  pixels region to the area tensor  $A^p$ . Let  $(x_0, y_0)$  be the position of the center of the lower left pixel, then  $x = x_0 + \Delta x$  and  $y = y_0 + \Delta y$  gives the values needed to compute the contribution via equation (2.26). If several entries are given for a single configuration, you need to apply (2.26) to each of them and sum the results to find the total contribution of the region. You need to divide the resulting contributions by 4 to account for doubly counted contributions (cf. table 2.3). The color code is the same as in table 2.3.

	$\Delta x$	$\Delta y$	$n_1$	$n_2$
				
	1/2	0	1	0
	0	1/2	0	1
	1/2	0	-1	0
	1	1/2	0	1
	0	1/2	0	1
	1	1/2	0	1
	1/2	1	1	0
	0	1/2	0	-1
	1/2	0	1	0
	1/2	1	1	0
	1/2	0	-1	0
	1/2	1	1	0
	0	1/2	0	-1
	1	1/2	0	1
	1	1/2	0	1
	1/2	1	1	0









	$\Delta x$	$\Delta y$	$n_1$	$n_2$
	1	1/2	0	-1
	1/2	1	-1	0
	1/2	0	1	0
	1/2	1	-1	0
	0	1/2	0	1
	1	1/2	0	-1
	1/2	0	-1	0
	1/2	1	-1	0
	0	1/2	0	1
	1/2	1	-1	0
	0	1/2	0	-1
	1	1/2	0	-1
	1/2	0	1	0
	1	1/2	0	-1
	1/2	0	-1	0
	0	1/2	0	-1
				

Table 2.6: Contributions of a  $2 \times 2$  pixels region to the perimeter tensor  $P^{p,q}$ .

Let  $(x_0, y_0)$  be the position of the center of the lower left pixel, then  $x = x_0 + \Delta x$  and  $y = y_0 + \Delta y$  gives the values needed to compute the contribution via equation (2.28). Entries with  $n_1 = 0$  correspond to horizontal perimeter sections, entries with  $n_2 = 0$  to vertical perimeter sections. If several entries are given for a single configuration, you need to apply (2.28) to each of them and sum the results to find the total contribution of the region. You need to divide the resulting contributions by 2 to account for doubly counted contributions (cf. table 2.3). The color code is the same as in table 2.3.

	$\chi$	$\Delta x$	$\Delta y$	$n_1$	$n_2$		$\chi$	$\Delta x$	$\Delta y$	$n_1$	$n_2$
	1/4	1/2	1/2	1	1		1/4	1/2	1/2	-1	-1
	1/4	1/2	1/2	-1	1		-1/4	1/2	1/2	-1	1
	1/4	1/2	1/2	1	-1		-1/4	1/2	1/2	1	-1
	-1/4	1/2	1/2	1	1		-1/4	1/2	1/2	-1	1
	-1/4	1/2	1/2	-1	-1		-1/4	1/2	1/2	1	-1
	-1/4	1/2	1/2	1	1		-1/4	1/2	1/2	-1	-1

Table 2.7: Contributions of a  $2 \times 2$  pixels region to the curvature tensor  $\chi^{p,q}$ .

Let  $(x_0, y_0)$  be the position of the center of the lower left pixel, then  $x = x_0 + \Delta x$  and  $y = y_0 + \Delta y$  gives the values needed to compute the contribution via equation (2.29). If several entries are given for a single configuration, you need to apply (2.29) to each of them and sum the results to find the total contribution of the region. The color code is the same as in table 2.3.

### Boundary Conditions

Just like with Minkowski functionals you need to take special care when evaluating boundary contributions of Minkowski tensors. While in principle you can apply the same techniques as described in 2.1.4, there are some additional caveats.

Consider e.g. figure 2.4, which shows a problem of periodic boundary conditions. While the Minkowski functionals of this image indicate single connected lines (bodies with  $A = 3$ ,  $P = 8$  and  $\chi = 1$ ) just as intended, the wrapping of coordinates distorts higher rank tensors and makes them indicate 2 separate bodies when an object extends beyond the glue lines. While these inconsistencies might be resolved by extending the framework of Minkowski tensors to manifolds (Alesker, 2007) and treating planar pixel images with periodic boundary conditions as pixels on a torus, there is no easy way to resolve them in the planar theory discussed in this work.

This does not necessarily render periodic boundary conditions utterly useless. If the number of boundary pixels is small compared to the total number of

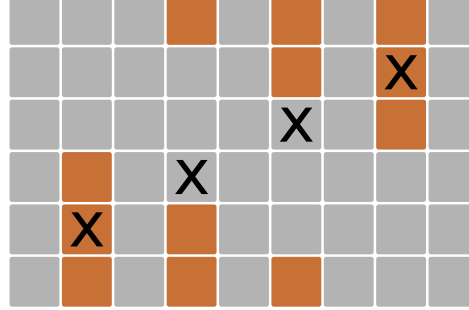


Figure 2.4: Schematic view of a 3 pixel line being wrapped around by periodic boundary conditions. The different pixel columns show the same body in different stages of sliding downwards. The center of mass of every brown line is in the center of the middle pixel. The center of mass inferred from the Minkowski tensor  $A^1$  is marked as black cross.

pixels, the Minkowski tensors will contain all relevant information of the bulk of the pixels. So, if the required accuracy is not too high, the inconsistencies and boundary artifacts may just be neglected. Nevertheless, you need to carefully evaluate the influence of periodic boundary conditions on your use case before applying them.

The biggest problem of minus sampling boundary conditions are the partial bodies, which might appear at the boundaries (cf. section 2.1.4). Besides yielding fractional contributions to the Euler characteristic, which introduces special information about a few boundary pixels into the fractional part of an otherwise integer valued quantity, these partial bodies change the transformation properties of some Minkowski tensors. Usually all rank 1 tensors based on normal vectors, i.e. the tensors  $M_{\nu}^{0,1}$ , are 0 (Müller, 1953). This makes the tensors  $M_{\nu}^{1,1}$  translation-invariant (cf. equation (2.25)). Figuratively speaking walking along a closed curve and always pointing outwards leads to pointing in all directions and so the  $M_{\nu}^{0,1}$  along the closed surface of a body are 0. The partial bodies introduced by minus sampling boundary conditions contribute only a part of their surface and thus provide no closed curve. With  $M_{\nu}^{0,1}$  not vanishing, the tensors  $M_{\nu}^{1,1}$  are not translation-invariant. So, quantities which usually do not depend on the chosen origin will yield different values



for shifted origins when using minus sampling boundary conditions.

Just like with periodic boundary conditions, the boundary effects of minus sampling boundary conditions may well be negligible for some applications, but you have to carefully evaluate the influence of the boundary conditions on your use case before applying them. The only alternative without negative impact on the underlying theory seems to be embedding the b/w image in a white environment. Setting all pixels beyond the given image to white closely reassembles the initial requirement of compact closed sets. While this prevents inconsistencies and allows for a smooth processing of the image, it may distort the inherent structure of the shown objects, as it adds surface perimeter sections to every black boundary pixel without accounting for the real environment beyond the given image, if the image provides just a clipped view of a larger system.

The contributions of white boundary conditions are identical to the contributions of interior pixels and may easily be computed by adding a row of white pixels to every side of the image and processing the usual  $2 \times 2$  regions of this enlarged image. The relevant contributions for minus sampling and periodic boundary conditions can easily be constructed from the details given in section 2.1.4 and the general structure of the contributions of internal pixels to the Minkowski tensors. Therefore, they are not listed here.

### Marching Squares Algorithm

As there are only 4 different perimeter normals in an image of square pixels, the resolution of tensors based on normal vectors is coarse. This is not a general problem of the affected Minkowski tensors though, but merely shows the significant impact of pixelation and binning errors on the distribution of normal vectors. To reduce these effects a so called marching square algorithm can be used. The basic idea of this algorithm is introducing diagonal lines into the square grid in order to reduce its anisotropy. Similar approaches are widely used in digital image processing and an elaborate variant of the marching square algorithm specially designed for Minkowski functionals of 2-dimensional gray scale images is discussed in Mantz et al. (2008).

In its most basic variant the algorithm assigns each of the 16 possible configurations of a  $2 \times 2$  pixels region a certain configuration of perimeter segments, as shown in figure 2.5.

These perimeter segments enclose a new polyconvex body, which is topo-

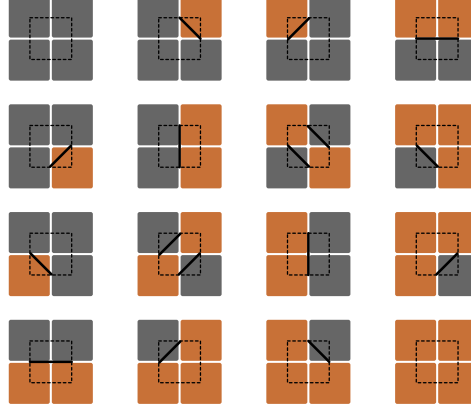


Figure 2.5: Mapping of the 16 different  $2 \times 2$  pixels configurations to marching square perimeter segments. The interior of the original pixel body is highlighted brown. The interior of the new marching square body is on the side of the shown perimeter section corresponding to the interior of the brown body.

logically equivalent to the polyconvex body of the original b/w pixel image (cf. figure 2.6). Connected regions are more evident in this new body and step-shaped boundaries are smoothed. The underlying structure of this new image is essentially based on triangular pixels (cf. figure 2.7) and the marching square algorithm gives instructions on which triangular pixels to paint black based on a given square pixel image.

Everything discussed so far can easily be extended to triangular pixels and local contributions of a marching square body. The biggest difference being that you have to integrate triangles instead of squares and you have to consider  $3 \times 3$  regions of square pixels instead of  $2 \times 2$  regions, as the corners of the marching square body are only visible when combining two of the configurations of figure 2.5 and the angle between adjacent perimeter segments determines the contribution to  $\chi^{p,q}$  (Schröder-Turk et al., 2010).

While the marching square algorithm allows to reduce pixelation errors, it is not used in this work and only mentioned for completeness, as it is very useful for a variety of structure studies in physics (Mantz et al., 2008). The aim of this work is not to study the underlying structure of a gray scale image

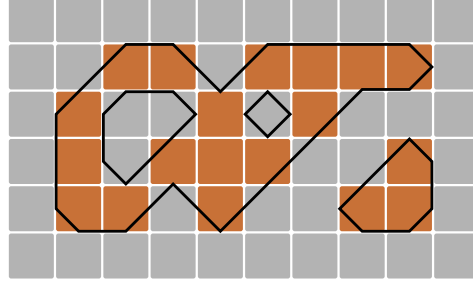


Figure 2.6: Example of a conversion from a square pixel body to a marching square body. The pixels of the original pixel body are highlighted brown. The black contours show the boundaries of the resulting marching square body.

directly, but to compare the found morphology to the expected morphology of a background image. As long as both suffer from the same binning effects, the structure of the background image may still be used as reference structure for observed images and the additional complexity of the marching square algorithm is not needed.

Actually this comparison of a measured gamma-ray counts map to the

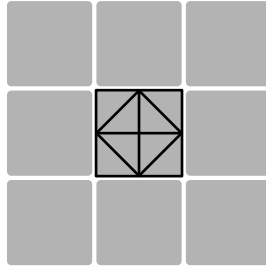


Figure 2.7: Partitioning of a square pixel into congruent triangles, as used by the marching square algorithm. The four central triangles are always colored like the enclosing square pixel by the algorithm, the corner triangles are colored depending on the surrounding pixels (cf. figure 2.5).

expected structure of such a map is not quite as straightforward as you might think. It is not sufficient to compare the Minkowski tensors of the measured image to the expectation value of the tensors given the background model of homogeneous and isotropic Poisson events (cf. equation (1.1)). As the probability distributions of Minkowski tensors for this background model are quite asymmetric, using only the first moment of these distributions, i.e. the expectation value, makes it very hard to uniquely quantify deviations from the expected value. In order to determine the significance of structural deviations of a measured image from a typical background image, you need to consider the full probability distributions of the involved tensors. How to determine these distributions is discussed in the following sections.

## 3 Computing Distributions of Minkowski Tensors

It is vital for a rigorous quantitative analysis of structures to know the probability distributions of the used valuations. The brute force approach of evaluating every possible configuration and counting the number of equivalent configurations can be used to infer distributions for small problem sizes, but the computational effort quickly outgrows the capabilities of modern hardware. The number of possible configurations of a b/w image of  $N \times N$  pixels is  $2^{N^2}$ . Thus, evaluating all configurations in a brute force approach leads to a super-exponential increase in the run-time with increasing number of pixels.

This chapter shows how to construct the desired distributions in sub-exponential run-time using methods from combinatorics and number theory. The first section provides a short introduction to the number theoretic background. It is shown how to calculate the number of partitions of a positive integer  $n$ , i.e. the number of ways to write  $n$  as a sum of positive integers. This is generalized in section 3.2 to counting the number of b/w pixel images which yield the same values for a given set of area tensors. Section 3.3 shows how to adjust the introduced formalism in order to incorporate functionals whose local contribution can not be determined by a single pixel, i.e. perimeter and Euler characteristic.

### 3.1 Partition Numbers

As the theory of partition numbers is not so commonly known among gamma-ray astrophysicists, this section gives a short introduction. The overview provided here is a short summary of the corresponding sections of Andrews (1998), focusing on the topics needed to calculate distributions of Minkowski tensors. For an in-depth discussion of generating functions and more details on partition numbers the interested reader is referred to Andrews (1998).

A partition of a positive integer  $n$  is a sorted tuple  $(\lambda_1, \lambda_2, \dots, \lambda_k)$  of positive integers  $\lambda_i$  for which holds:

$$i < j \Rightarrow \lambda_i \geq \lambda_j \quad \wedge \quad \sum_{i=1}^k \lambda_i = n. \quad (3.1)$$

As the order of the summands is not important, it is often convenient to write the partition in a different way, showing only distinct summands and the number of times each summand occurred in the partition:  $(\rho_1^{f_1}, \rho_2^{f_2}, \dots, \rho_s^{f_s})$ , where  $i < j \Leftrightarrow \rho_i > \rho_j$  and  $f_i$  is the number of occurrences of the summand  $\rho_i$  in the partition.

The theory of partitions is concerned with the number of possible partitions of  $n$  given certain constraints for  $\lambda_i$ . It turns out that the partition function  $p(n)$ , which maps  $n$  to the number of possible partitions of  $n$ , can be constructed quite easily via a generating function

$$f(q) = \sum_{n=0}^{\infty} p(n)q^n. \quad (3.2)$$

Instead of discussing the motivation for a generating function of this form and deriving general rules to construct the generating function for a certain problem, some examples of generating functions and their properties are given. This should be enough to get an intuition on how to build and how to use them. Further details — like the conditions needed for  $f(q)$  to be well defined and how to calculate  $p(n)$  if  $f(q)$  is not a simple polynomial in  $q$  — are given in Andrews (1998).

Given a set of possible summands  $\mathcal{R}$  and limiting the multiplicity  $f_\rho$  of every summand  $\rho \in \mathcal{R}$  to  $d$  the generating function is given by

$$f(q) = \prod_{\rho \in \mathcal{R}} \sum_{f_\rho=0}^d q^{\rho f_\rho}. \quad (3.3)$$

Expanding the product via the distributive law yields one summand for every possible combination of summands  $\rho$  and multiplicities  $f_\rho$ . In the expanded formula the power of  $q$  in every term corresponds to the sum of the  $\rho$ s with the corresponding multiplicities  $f_\rho$ . Simplifying the formula by collecting all terms of equal powers in  $q$  results in coefficients which correspond to the number of partitions of the corresponding exponent.

While the powers of  $q$  are designed to keep track of the contributions of different summands and their sum, additional variables can be added to the generating function to draw additional information from the generating function. The function

$$f(q, t) = \prod_{\rho \in \mathcal{R}} \sum_{f_\rho=0}^d q^{\rho f_\rho} t^{f_\rho} \quad (3.4)$$

is a generating function for the number of partitions  $p(n, k)$  of  $n$  with a certain number of summands  $k$ , i.e.

$$f(q, t) = \sum_{n=0}^{\infty} \sum_{k=0}^{\infty} p(n, k) q^n t^k, \quad (3.5)$$

as  $t$  keeps track of the number of summands that contributed to the sum, which in turn is tracked by  $q$ .

While  $\mathcal{R}$  is usually a subset of  $\mathbb{N}$  in mathematical considerations, the principle of marking additive values by variables like  $q$  or  $t$  and using polynomial multiplication to count the number of equivalent configurations is very universal and can be quite helpful when calculating distributions of Minkowski tensors. In this case the elements of  $\mathcal{R}$  are pixels of a b/w image and the powers of the variables of the generating function are designed to correspond to the contributions of the pixel to Minkowski tensors. The details of this procedure are discussed in the next section.

## 3.2 Distributions of the Area Tensors

The area functional  $A$  is the only functional where the probability distribution can be written down immediately, without the need for any heavy framework. Given the probability  $p$  of a pixel to be black and assuming uncorrelated pixels, the number of black pixels in an  $N \times N$  b/w image is binomially distributed. As  $A$  is just the count of black pixels, one finds

$$\mathcal{P}(A) = \binom{N^2}{A} p^A (1-p)^{N^2-A}. \quad (3.6)$$

The number of configurations for a certain  $A$  is just given by the binomial coefficient  $\binom{N^2}{A}$ .

First, this result is reproduced using generating functions as introduced in section 3.1, then this ansatz is generalized to count the number of configurations of a certain area tensor. At the end of this section it is shown how to determine the common probability distribution of an arbitrary set of area tensors using the developed techniques.

Let  $\mathcal{N}(A; N)$  be the number of  $N \times N$  b/w images with  $A$  black pixels. As  $\mathcal{N}(A; N) = \binom{N^2}{A}$  is already known, the generating function for  $\mathcal{N}(A; N)$  is

$$\begin{aligned}
 f(q) &= \sum_{a=0}^{\infty} \mathcal{N}(a; N) q^a \\
 &= \sum_{a=0}^{N^2} \binom{N^2}{a} q^a \\
 &= (1 + q)^{N^2} \\
 &= \prod_{i=1}^{N^2} (1 + q).
 \end{aligned} \tag{3.7}$$

Comparing this with equation (3.3) allows the following interpretation:

The generating function is a product over all items which might contribute to the functional, i.e. a product over all pixels of the image. The factors of the product are sums, where each summand corresponds to a possible contribution of the item, i.e. every pixel can contribute a value of 0 (white pixel) or 1 (black pixel) to the final area, which corresponds to the factors  $(q^0 + q^1)$ .

Starting from this point of view, it is possible to construct generating functions for the number of configurations of area tensors. Before giving the general formula, this is demonstrated for the area vector.

As the area vector has two independent components, one needs two variables to mark the corresponding term in the generating function. Let  $a_x$  and  $a_y$  be the components of the area vector. The generating function for the number of  $N \times N$  b/w images  $\mathcal{N}(a_x, a_y; N)$  of a given area vector  $(a_x, a_y)$  is given by

$$f(q_x, q_y) = \sum_{a_x=0}^{\infty} \sum_{a_y=0}^{\infty} \mathcal{N}(a_x, a_y; N) q_x^{a_x} q_y^{a_y}. \tag{3.8}$$



To construct the generating function, one needs to find all possible contributions of a certain pixel, build a corresponding sum and finally compute the product of all pixel terms. As before, there are only two possible contributions for every pixel, 0 for white pixels and some contribution depending on the coordinates of the pixel, if the pixel is black (cf. section 2.2.3). Let  $x$  and  $y$  be the coordinates of the pixel, then the contribution to  $a_x$  is  $x$  and the contribution to  $a_y$  is  $y$ . This leads to the term  $(1 + q_x^x q_y^y)$  and to the generating function

$$f(q_x, q_y) = \prod_{x=1}^N \prod_{y=1}^N (1 + q_x^x q_y^y). \quad (3.9)$$

In general the generating function of the number of image configurations for a certain value of an area tensor can be computed via

$$f(q_1, q_2, \dots, q_s) = \prod_{x=1}^N \prod_{y=1}^N \left( 1 + \prod_{i=1}^s q_i^{w_i(x,y)} \right), \quad (3.10)$$

where each of the  $q_i$  corresponds to one of the components of the tensor and the weight functions  $w_i(x, y)$  determine the contribution of a black pixel at the given position to the value of this tensor component.

It is straightforward to generalize equation (3.10) to a combination of tensors, i.e. to construct a generating function for the number of images which have equal values for all chosen area tensors:

$$f(\{q_{ij}\}) = \prod_{x=1}^N \prod_{y=1}^N \left( 1 + \prod_{j=1}^k \prod_{i=1}^{s_j} q_{ij}^{w_{ij}(x,y)} \right), \quad (3.11)$$

where  $j$  enumerates the different tensors and  $i$  the different components of each tensor;  $\{q_{ij}\}$  is just a shortened notation for all the necessary variables,  $k$  the number of tensors which are considered simultaneously and  $s_j$  the number of components of the  $j^{\text{th}}$  tensor.

The common probability distribution of a set of tensors is related to the just derived number of equivalent configurations in the same way as the binomial distribution to the binomial coefficients. The probability to find one specific image of certain tensor values has to be multiplied with the just derived number of all equivalent configurations. Given that the pixels are independent, the probability of a certain b/w image depends only on the

number of black pixels, i.e. on the area  $A$ . Just like in equation (3.6) the probability of a certain image is given by  $p^A(1-p)^{N^2-A}$ . Let  $\mathcal{T}$  be the set of area tensors to be considered and  $\mathcal{N}_{\mathcal{T}}(\{a_{ij}\}; N)$  the number of  $N \times N$  b/w images which lead to the same values for all the components  $\{a_{ij}\}$  of the tensors in  $\mathcal{T}$ , where  $i$  enumerates the different components of the  $j^{\text{th}}$  tensor and  $j$  the tensors in  $\mathcal{T}$ . If the area functional  $A$  (as tensor of rank 0) is in  $\mathcal{T}$ , let  $a_{11}$  be the component from  $\{a_{ij}\}$  which corresponds to  $A$ , i.e.  $a_{11} := A$ . The probability to find a given set of tensor values is then given by

$$\mathcal{P}_{\mathcal{T}}(\{a_{ij}\}) = \mathcal{N}_{\mathcal{T}}(\{a_{ij}\}; N) p^{a_{11}}(1-p)^{N^2-a_{11}}. \quad (3.12a)$$

If  $A \notin \mathcal{T}$ , let  $\mathcal{T}^* := \mathcal{T} \cup \{A\}$ , then the probability to find a given set of tensor values is given by

$$\mathcal{P}_{\mathcal{T}}(\{a_{kl}\}) = \sum_{a_{11}=0}^{N^2} \mathcal{N}_{\mathcal{T}^*}(\{a_{ij}\}; N) p^{a_{11}}(1-p)^{N^2-a_{11}}, \quad (3.12b)$$

where the components  $\{a_{kl}\}$  are indexed in a way excluding the case  $k = l = 1$ .

If the full probability distribution  $\mathcal{P}_{\mathcal{T}}(\{a_{ij}\})$  is needed, all coefficients of the generating function  $f(\{q_{ij}\})$  are needed, as they are closely linked to the probability distribution via equation (3.12). A first hint on the efficiency of this approach compared to the brute force algorithm can be taken from equation (3.11). The needed coefficients are calculated by successively adding one factor per pixel to the product, while the brute force approach has to generate all possible configurations and to reevaluate the contribution of every pixel for each image. This usually leads to a huge performance boost and allows the calculation of distributions for problems, where the brute force approach would take several life times to finish. A detailed performance comparison is given later in section 3.4.

### 3.3 Distributions of Minkowski Tensors

The methods developed so far can not be applied directly to functionals like the perimeter  $P$  or the Euler characteristic  $\chi$ . While the contribution to any component of an area tensor can be reduced to the contribution of independent pixels, the contributions to  $P$ ,  $\chi$  and their respective families of tensors depend on several pixels and can not be computed independently. As

discussed in chapter 2, these correlations between pixels lead to a need for boundary conditions to determine the value of pixels in absence of neighboring pixels (cf. section 2.1.4), but for interior pixels it is still possible to use the additivity of the Minkowski tensors to calculate local contributions of  $2 \times 2$  pixel clusters and to get the final value of the tensors by summing these local contributions (cf. section 2.2.3).

In the course of this section techniques are developed to include correlations between pixels in the construction of a generating function of the number of  $N \times N$  b/w images with identical tensor values. Once this function is constructed a formula very similar to equation (3.12) can be used to compute the probability distribution from the coefficients of the generating function.

### 3.3.1 Contribution of Interior Pixels

The final algorithm starts at the lower left corner pixel and work its way up row by row to the upper right corner. However, starting with boundary pixels and the influence of boundary conditions on the pixel contributions distracts from the general structure of the algorithm, therefore, this section discusses the contribution of interior pixels and boundary effects are discussed in section 3.3.2.

The generating function  $f(\{q_{ij}\})$  of the number of  $N \times N$  b/w images with identical tensor values can be constructed via a sequence  $f_{xy}(\{q_{ij}\}, \{b_{kl}, w_{kl}\})$ , which describes the contributions to the generating function up to the pixel at  $(x, y)$ . While the variables  $\{q_{ij}\}$  are tracking the values of different tensor components, like in equation (3.11), the variables  $\{b_{kl}, w_{kl}\}$  are used to track the correlations between pixels. The final generating function  $f(\{q_{ij}\}) = f_{NN}(\{q_{ij}\})$  does not depend on these additional helper variables. The sequence is defined by the following recurrence relation

$$f_{xy}(\{q_{ij}\}, \{b_{kl}, w_{kl}\}) = \hat{O}_{xy} f_{(x-1)y}(\{q_{ij}\}, \{b_{kl}, w_{kl}\}), \quad (3.13)$$

where the operator  $\hat{O}_{xy}$  adds the contribution of the current pixel to the previous sequence element. By setting  $\hat{O}_{xy} = \left(1 + \prod_{j=1}^k \prod_{i=1}^{s_j} q_{ij}^{\text{weight}_{ij}(x,y)}\right)$  the recurrence relation can be used to reproduce equation (3.11).<sup>1</sup> In order

---

<sup>1</sup>How to get to the next  $y$  when the boundary of  $x$  is reached is discussed in the context of boundary conditions in section 3.3.2.

to construct the more general operator needed for the current problem, it is useful to define the two operators

$$\begin{aligned}\hat{0}(x_k)f(\{x_i\}) &:= f(\{x_i\})|_{x_k=0} \\ \hat{1}(x_k)f(\{x_i\}) &:= f(\{x_i\})|_{x_k=1},\end{aligned}\tag{3.14}$$

which set a given variable to 0 or 1 respectively. It is important to obey the commutator relations of these operators

$$\begin{aligned}[\hat{0}(x_k), \hat{0}(x_i)] &= 0 \\ [\hat{1}(x_k), \hat{1}(x_i)] &= 0 \\ [\hat{1}(x_k), \hat{0}(x_i)] &= \delta_{ik} (\hat{0}(x_i) - \hat{1}(x_i)).\end{aligned}\tag{3.15}$$

when simplifying products of operators.

$\hat{O}_{xy}$  can be constructed by similar concepts as equation (3.11). The operators  $\hat{O}_{xy}$  in the recurrence relation correspond to the factors of equation (3.11). These factors are constructed from sums, where each summand corresponds to a possible contribution of the current pixel (cf. section 3.2). Following this scheme, the operator  $\hat{O}_{xy}$  is a sum of operators, which correspond to the possible contributions of pixel  $(x, y)$ :  $\hat{O}_{xy} = \sum_i \hat{X}_i$ . As discussed in section 2.2.3, these contributions can be mapped to the 16 possible configurations of a  $2 \times 2$  pixel region around the current pixel and without loss of generality one can assume that the current pixel is the upper right pixel of this  $2 \times 2$  region:  $\hat{O}_{xy} = \sum_{i=1}^{16} \hat{X}_i$ . The 16 possible configurations can be separated into 2 groups of 8 configurations each: One group where the current pixel is black and one, where it is white:  $\hat{O}_{xy} = \sum_{i=1}^8 \hat{B}_i + \hat{W}_i$ . Later it is important to be able to discriminate between contributions created by  $\hat{B}_i$  and contributions created by  $\hat{W}_i$ , as the state of this pixel influences the contribution of others, where this pixel is in their  $2 \times 2$  region. Therefore, the helper variables mentioned in the beginning of this section are used to tag the corresponding terms:  $\hat{O}_{xy} = \sum_{i=1}^8 b_{xy} \hat{B}'_i + w_{xy} \hat{W}'_i$ . If the same tagging scheme has been applied to the earlier elements of the sequence, the helper variables can be used to remove conflicting terms from the previous sequence element. E.g. to find the contribution of an all white  $2 \times 2$  region, the  $b_{xy}$  of all the pixels in the region can be set to 0 to remove wrong contributions to the all white case.

Combining all these ideas leads to the operator

$$\begin{aligned}
 \hat{O}'_{xy} &= b_{xy} \prod_{j=1}^k \prod_{i=1}^{s_j} q_{ij}^{\text{weight}_{ij}(\text{cfg}_1)} \hat{0}(w_{(x-1)(y)}) \hat{0}(w_{(x-1)(y-1)}) \hat{0}(w_{(x)(y-1)}) \\
 &+ w_{xy} \prod_{j=1}^k \prod_{i=1}^{s_j} q_{ij}^{\text{weight}_{ij}(\text{cfg}_2)} \hat{0}(w_{(x-1)(y)}) \hat{0}(w_{(x-1)(y-1)}) \hat{0}(w_{(x)(y-1)}) \\
 &+ \dots,
 \end{aligned} \tag{3.16}$$

where the different configurations  $\text{cfg}_i$  are determined by the color of the current pixel ( $b_{xy}/w_{xy}$ ) and the colors of the 3 remaining pixels in the  $2 \times 2$  region (as determined by the operators  $\hat{0}$ , which remove conflicting terms from the previous sequence element). The weight functions  $\text{weight}_{ij}(\text{cfg}_i)$  correspond to the  $w_{ij}$  in equation (3.11) and yield the contribution of the  $2 \times 2$  region to the  $i^{\text{th}}$  component of the  $j^{\text{th}}$  tensor (cf. tables 2.5, 2.6 and 2.7). All other variables are analogous to the corresponding variables in equation (3.11).

The first summand in equation (3.16) corresponds to the contribution of the all black configuration, as the current pixel is tagged black by  $b_{xy}$  and the white contributions of the other 3 pixels are removed. The other 15 summands are very similar, always tagging the current pixel with a certain color and removing contributions of the wrong color of the other 3 pixels.

The final operator  $\hat{O}_{xy}$  is given by

$$\hat{O}_{xy} = \hat{1}(b_{(x-1)(y-1)}) \hat{1}(w_{(x-1)(y-1)}) \hat{O}'_{xy}. \tag{3.17}$$

As the pixel at  $(x-1, y-1)$  does not influence further contributions the remaining tags for this pixel can be removed by setting the variables to 1.

Figure 3.1 depicts the different pixel classes involved in the calculation of  $f_{xy}$ . Brown pixels are currently tagged by helper variables. When calculating the contribution of the red frame,  $\hat{O}'_{xy}$  converts the unprocessed light gray pixel on the upper right of the red frame to a brown pixel by tagging it and adding the contribution of the red frame to the generating function. The final  $\hat{1}$  operators convert the brown pixel on the lower right to dark gray, as this pixel is too far away from unprocessed pixels to influence their contributions and thus all of its contributions that were not removed earlier using  $\hat{0}$  operators can fully contribute to the final function.

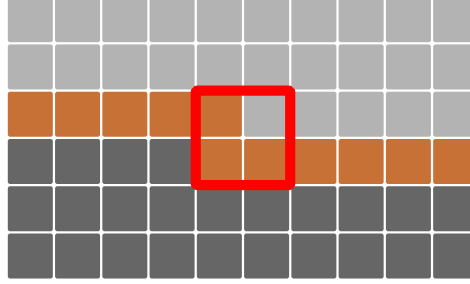


Figure 3.1: Schematic view of different pixel classes to consider when computing the contribution of interior pixels to the generating function of the number of  $N \times N$  b/w images with identical tensor values. Light gray pixels depict unprocessed pixels with unknown contributions. Dark gray pixels are fully processed and do not influence the contributions of further pixels. Brown pixels are in an intermediate state, which is described in detail in the text of this section. The red frame highlights the  $2 \times 2$  region which needs to be processed in order to convert the upper right pixel in the frame to brown and the lower left pixel to dark gray.

It is important to use  $\hat{O}_{xy}$  for an efficient algorithm and not  $\hat{O}'_{xy}$ . If unneeded tags are not removed, every operator  $\hat{O}_{xy}$  introduces 2 classes of summands, which can never be simplified, as the one class contains  $b_{xy}$  while the other class contains  $w_{xy}$ . This leads to  $2^{N \times N}$  different tag combinations for  $N \times N$  pixels, i.e.  $\mathcal{O}(2^{N^2})$  different summands in the generating function. Thus, an algorithm based on  $\hat{O}'_{xy}$  is not considerably more efficient than the brute force algorithm of processing all  $2^{N^2}$  configurations separately. Due to the non vanishing commutator relations from equation (3.15) the operators  $\hat{O}_{xy}$  do not commute for different  $x$  and  $y$ , as  $\hat{O}_{xy}$  comprises as well  $\hat{o}$  operators to remove conflicting terms as  $\hat{l}$  operators to reduce the complexity of the expression and simplify the resulting polynomial. Therefore, an algorithm based on  $\hat{O}_{xy}$  is very sequential; every element of the sequence  $f_{xy}$  depends on the previous element of the sequence and reordering of operations is very limited. Still, the early removal of tag variables allows simplifications which lead to a significant reduction of complexity in the calculation of the generating function. Details

on the performance of the algorithm are given in section 3.4.

### 3.3.2 Contribution of Boundary Pixels

While the essential points of the algorithm have been discussed in the previous section, there are still some open questions:

- What are the initial values for the recurrence relation in equation (3.13)?
- Equation (3.13) specifies how to include the next pixel in a given row  $y$ , but not how to proceed to the next row. How are the end of the current row and the start of a new row handled?
- How are the tagging variables of the final row removed, in order to obtain a tag free final generating function?

All of these questions are related to the chosen boundary conditions. Instead of developing a general, abstract framework to include arbitrary boundary conditions, this section gives two specific examples. It should be straightforward to adopt the given methods to other boundary conditions.

#### Minus Sampling Boundary Conditions

The first example of boundary contributions discusses minus sampling boundary conditions (cf. section 2.1.4). Figure 3.2 depicts the contribution of the first pixel, using these boundary conditions. As the outside pixel on the lower left is not going to contribute to any other  $2 \times 2$  region, there is no need to tag it with helper variables. The other pixels are tagged and the 16 possible configurations of this first region provides the initial term of the sequence:

$$\begin{aligned}
 f_{11} = & b_{11} b_{01} b_{10} \prod_{j=1}^k \prod_{i=1}^{s_j} q_{ij}^{\text{weight}_{ij}(\text{cfg}_1)} \\
 & + w_{11} b_{01} b_{10} \prod_{j=1}^k \prod_{i=1}^{s_j} q_{ij}^{\text{weight}_{ij}(\text{cfg}_2)} \\
 & + \dots
 \end{aligned} \tag{3.18}$$

The terms in this equation correspond to equivalent terms in equation (3.16), with the difference that there are no previously tagged pixels in the

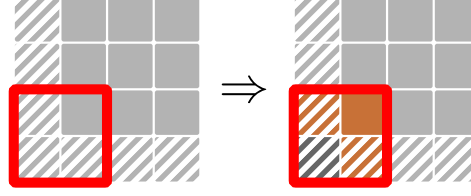


Figure 3.2: Schematic view of the contribution of the first pixel in the first row of pixels to the generating function of the number of equivalent tensor configurations using minus sampling boundary conditions. Hatched pixels lie on the outside of the  $N \times N$  pixels region and are used to determine the contribution of boundary pixels. The left hand side shows the state of the pixels before processing the red region, the right hand side afterwards. The color code is the same as in figure 3.1.

contribution region and there is no need to remove conflicting contributions from previous regions. Thus, instead of  $\hat{0}$  operators there are additional tag variables for all unprocessed pixels. Where all the terms with white tags are removed in equation (3.16), there is an explicit black tag for the corresponding pixel in equation (3.18) and vice versa. The tags for the pixel in the lower left corner can immediately be set to 1, as this pixel does not contribute to any other  $2 \times 2$  region.

The initial sequence item  $f_{11}$  is *not* the initial value of the recurrence relation in equation (3.13) with the operator given in equation (3.17). The entire first row involves boundary pixels and thus needs special treatment. Figure 3.3 depicts the contribution of pixels in the first row of the  $N \times N$  region. As before, unprocessed pixels are tagged; already tagged pixels are treated like in the case of interior pixels: conflicting terms are removed using  $\hat{0}$  operators and tags of pixels which do not contribute to further contribution regions are removed using  $\hat{1}$  operators. This leads to the following operator of the first row:



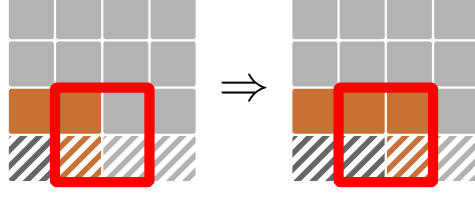


Figure 3.3: Schematic view of the contribution of the first row of pixels to the generating function of the number of equivalent tensor configurations using minus sampling boundary conditions. Hatched pixels lie on the outside of the  $N \times N$  pixels region and are used to determine the contribution of boundary pixels. The left hand side shows the state of the pixels before processing the red region, the right hand side afterwards. The color code is the same as in figure 3.1.

$$\begin{aligned} \hat{O}'_{x1} &= b_{x1} b_{x0} \prod_{j=1}^k \prod_{i=1}^{s_j} q_{ij}^{\text{weight}_{ij}(\text{cfg}_1)} \hat{0}(w_{(x-1)(1)}) \hat{0}(w_{(x-1)(0)}) \\ &\quad + w_{x1} b_{x0} \prod_{j=1}^k \prod_{i=1}^{s_j} q_{ij}^{\text{weight}_{ij}(\text{cfg}_2)} \hat{0}(w_{(x-1)(1)}) \hat{0}(w_{(x-1)(0)}) \end{aligned} \quad (3.19)$$

+ ...

$$\hat{O}_{x1} = \hat{1}(b_{(x-1)(0)}) \hat{1}(w_{(x-1)(0)}) \hat{O}'_{x1} \quad (3.20)$$

Plugging this operator in the recurrence relation (3.13) properly handles the contributions of the pixels of the first row.

The first row is finalized by the contribution of the lower right corner pixel, as shown in figure 3.4. Following the same principles as before, the operator describing the contribution of the lower right corner is given by:

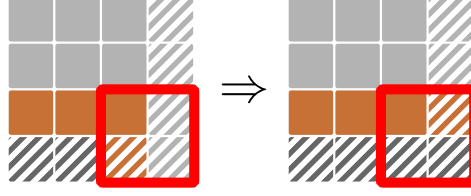


Figure 3.4: Schematic view of the contribution of the last pixel in the first row of pixels to the generating function of the number of equivalent tensor configurations using minus sampling boundary conditions. Hatched pixels lie on the outside of the  $N \times N$  pixels region and are used to determine the contribution of boundary pixels. The left hand side shows the state of the pixels before processing the red region, the right hand side afterwards. The color code is the same as in figure 3.1.

$$\begin{aligned} \hat{O}'_{(N+1)(1)} &= b_{(N+1)(1)} \prod_{j=1}^k \prod_{i=1}^{s_j} q_{ij}^{\text{weight}_{ij}(\text{cfg}_1)} \hat{O}(w_{N1}) \hat{O}(w_{N0}) \\ &+ w_{(N+1)(1)} \prod_{j=1}^k \prod_{i=1}^{s_j} q_{ij}^{\text{weight}_{ij}(\text{cfg}_2)} \hat{O}(w_{N1}) \hat{O}(w_{N0}) \\ &+ \dots \end{aligned} \quad (3.21)$$

$$\hat{O}_{(N+1)(1)} = \hat{1}(b_{N0}) \hat{1}(w_{N0}) \hat{O}'_{(N+1)1} \quad (3.22)$$

So, the operator of the lower right corner is very similar to the operator of the other pixels in the first row, but the tags of the lower right pixel in the contribution region can be omitted, as this pixel does not contribute to any other region.

Figure 3.5 shows how to open the next row. This operator can be applied after processing the lower right corner to start the processing of the second row, and the same operator also opens the third or any other row, after finalizing the previous row of pixels. The given scheme corresponds to the following opening operator:



Figure 3.5: Schematic view of the contribution of the first pixel in the next row of pixels to the generating function of the number of equivalent tensor configurations using minus sampling boundary conditions. Hatched pixels lie on the outside of the  $N \times N$  pixels region and are used to determine the contribution of boundary pixels. The left hand side shows the state of the pixels before processing the red region, the right hand side afterwards. The color code is the same as in figure 3.1.

$$\begin{aligned} \hat{O}'_{1y} &= b_{1y} b_{0y} \prod_{j=1}^k \prod_{i=1}^{s_j} q_{ij}^{\text{weight}_{ij}(\text{cfg}_1)} \hat{0}(w_{(0)(y-1)}) \hat{0}(w_{(1)(y-1)}) \\ &\quad + w_{1y} b_{0y} \prod_{j=1}^k \prod_{i=1}^{s_j} q_{ij}^{\text{weight}_{ij}(\text{cfg}_2)} \hat{0}(w_{(0)(y-1)}) \hat{0}(w_{(1)(y-1)}) \\ &\quad + \dots \end{aligned} \tag{3.23}$$

$$\hat{O}_{1y} = \hat{1}(b_{(0)(y-1)}) \hat{1}(w_{(0)(y-1)}) \hat{O}'_{1y} \tag{3.24}$$

All rows opened by this operator can be processed by the interior pixel operator of the last section (cf. equation (3.17)).

The scheme to finalize a row when the last pixel is reached is given in figure 3.6. Its corresponding finalization operator is given by:

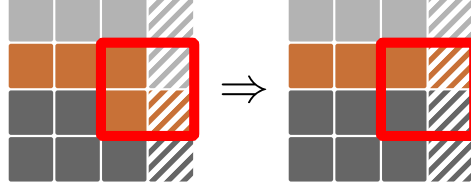


Figure 3.6: Schematic view of the contribution of the last pixel in the current row of pixels to the generating function of the number of equivalent tensor configurations using minus sampling boundary conditions. Hatched pixels lie on the outside of the  $N \times N$  pixels region and are used to determine the contribution of boundary pixels. The left hand side shows the state of the pixels before processing the red region, the right hand side afterwards. The color code is the same as in figure 3.1.

$$\begin{aligned}
 \hat{O}'_{(N+1)(y)} &= b_{(N+1)(y)} \prod_{j=1}^k \prod_{i=1}^{s_j} q_{ij}^{\text{weight}_{ij}(\text{cfg}_1)} \hat{o}(w_{Ny}) \hat{o}(w_{(N)(y-1)}) \hat{o}(w_{(N+1)(y-1)}) \\
 &\quad + w_{(N+1)(y)} \prod_{j=1}^k \prod_{i=1}^{s_j} q_{ij}^{\text{weight}_{ij}(\text{cfg}_2)} \hat{o}(w_{Ny}) \hat{o}(w_{(N)(y-1)}) \hat{o}(w_{(N+1)(y-1)}) \\
 &\quad + \dots
 \end{aligned} \tag{3.25}$$

$$\hat{O}_{(N+1)(y)} = \hat{1}(b_{(N)(y-1)}) \hat{1}(w_{(N)(y-1)}) \hat{1}(b_{(N+1)(y-1)}) \hat{1}(w_{(N+1)(y-1)}) \hat{O}'_{(N+1)y} \tag{3.26}$$

Repeated application of opening operator, interior pixel contributions and finalization operator can be used to process all rows of the  $N \times N$  pixel region up to the last row. Due to the additional boundary on the top of the last row, the opening operator has to be adjusted slightly to start the processing of this final row. Figure 3.7 shows how to open the last row of pixels. The corresponding opening operator is given by:

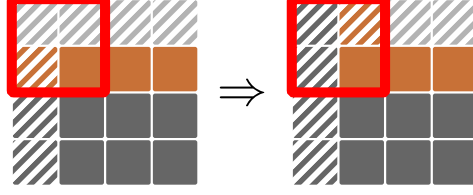


Figure 3.7: Schematic view of the contribution of the first pixel in the last row of pixels to the generating function of the number of equivalent tensor configurations using minus sampling boundary conditions. Hatched pixels lie on the outside of the  $N \times N$  pixels region and are used to determine the contribution of boundary pixels. The left hand side shows the state of the pixels before processing the red region, the right hand side afterwards. The color code is the same as in figure 3.1.

$$\begin{aligned} \hat{O}'_{(1)(N+1)} &= b_{(1)(N+1)} \prod_{j=1}^k \prod_{i=1}^{s_j} q_{ij}^{\text{weight}_{ij}(\text{cfg}_1)} \hat{O}(w_{0N}) \hat{O}(w_{1N}) \\ &+ w_{(1)(N+1)} \prod_{j=1}^k \prod_{i=1}^{s_j} q_{ij}^{\text{weight}_{ij}(\text{cfg}_2)} \hat{O}(w_{0N}) \hat{O}(w_{1N}) \\ &+ \dots \end{aligned} \quad (3.27)$$

$$\hat{O}_{(1)(N+1)} = \hat{1}(b_{0N}) \hat{1}(w_{0N}) \hat{O}'_{(1)(N+1)} \quad (3.28)$$

Tags for the upper left pixel can be omitted, as this pixel does not contribute to any other region.

The scheme for processing the final row of pixels is shown in figure 3.8. This results in an operator which reduces the number of tagged pixels step by step:

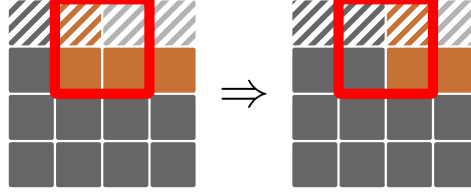


Figure 3.8: Schematic view of the contribution of the last row of pixels to the generating function of the number of equivalent tensor configurations using minus sampling boundary conditions. Hatched pixels lie on the outside of the  $N \times N$  pixels region and are used to determine the contribution of boundary pixels. The left hand side shows the state of the pixels before processing the red region, the right hand side afterwards. The color code is the same as in figure 3.1.

$$\begin{aligned}
 \hat{O}'_{(x)(N+1)} &= b_{(x)(N+1)} \prod_{j=1}^k \prod_{i=1}^{s_j} q_{ij}^{\text{weight}_{ij}(\text{cfg}_1)} \hat{o}(w_{(x-1)(N)}) \hat{o}(w_{xN}) \hat{o}(w_{(x-1)(N+1)}) \\
 &+ w_{(x)(N+1)} \prod_{j=1}^k \prod_{i=1}^{s_j} q_{ij}^{\text{weight}_{ij}(\text{cfg}_2)} \hat{o}(w_{(x-1)(N)}) \hat{o}(w_{xN}) \hat{o}(w_{(x-1)(N+1)}) \\
 &+ \dots
 \end{aligned} \tag{3.29}$$

$$\hat{O}_{(x)(N+1)} = \hat{1}(b_{(x-1)(N)}) \hat{1}(w_{(x-1)(N)}) \hat{1}(b_{(x-1)(N+1)}) \hat{1}(w_{(x-1)(N+1)}) \hat{O}'_{(x)(N+1)} \tag{3.30}$$

The final contribution of the upper right corner removes the remaining tag variables and provide the desired generating function. A schematic view of this last operation is given in figure 3.9 and the corresponding operator is given by:

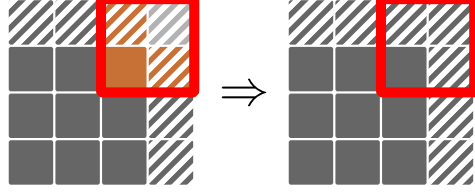


Figure 3.9: Schematic view of the contribution of the last pixel in the last row of pixels to the generating function of the number of equivalent tensor configurations using minus sampling boundary conditions. Hatched pixels lie on the outside of the  $N \times N$  pixels region and are used to determine the contribution of boundary pixels. The left hand side shows the state of the pixels before processing the red region, the right hand side afterwards. The color code is the same as in figure 3.1.

$$\begin{aligned} \hat{O}'_{(N+1)(N+1)} &= \prod_{j=1}^k \prod_{i=1}^{s_j} q_{ij}^{\text{weight}_{ij}(\text{cfg}_1)} \hat{O}(w_{NN}) \hat{O}(w_{(N+1)(N)}) \hat{O}(w_{(N)(N+1)}) \\ &+ \prod_{j=1}^k \prod_{i=1}^{s_j} q_{ij}^{\text{weight}_{ij}(\text{cfg}_2)} \hat{O}(w_{NN}) \hat{O}(w_{(N+1)(N)}) \hat{O}(w_{(N)(N+1)}) \\ &+ \dots \end{aligned} \quad (3.31)$$

$$\begin{aligned} \hat{O}_{(N+1)(N+1)} &= \hat{1}(b_{NN}) \hat{1}(w_{NN}) \\ &\quad \hat{1}(b_{(N)(N+1)}) \hat{1}(w_{(N)(N+1)}) \\ &\quad \hat{1}(b_{(N+1)(N)}) \hat{1}(w_{(N+1)(N)}) \hat{O}'_{(N+1)(N+1)} \end{aligned} \quad (3.32)$$

While this example was motivated by minus sampling boundary conditions, the formalism discussed so far can be applied to a much broader spectrum of boundary conditions. Actually all boundary conditions which determine the state of exterior pixels independently from the pixels in the  $N \times N$  region can be processed in the shown way. Besides minus sampling boundary conditions, methods like setting all exterior pixels to black, white or a check pattern can for example be processed the same way. The differences manifest in the actual contributions of the different boundary configurations, i.e. in  $\text{weight}_{ij}(\text{cfg}_k)$ .

The weights are not universal, e.g.  $\text{weight}_{ij}(\text{cfg}_k)$  may have different values in the first row compared to the formulas for interior pixels, which in turn might be different from the weight functions in the opening operator of the current line. The proper values of the weight functions have to be determined according to the used tensors and boundary conditions. How to find the correct contribution of a  $2 \times 2$  region is discussed in section 2.2.3.

#### Periodic Boundary Conditions

While the previous example was discussed in quite some detail, the formulas of the different operators are not explicitly stated for all configurations in the second example. Instead the general concepts that were used so far to translate the drawings to formulas is summarized. Together with the explicit formulas of the previous example this should provide enough insight to allow the construction of all needed operators from schematic drawings.

The rules to translate a schematic view of the contribution to the mathematical operator correspond to the color transitions of the participating pixels in the way described in table 3.1. By revisiting the previous example with this set of rules, the connection between the schematic drawings and the operator formulas should be quite clear.

Figure 3.10 depicts the contributions of the first pixel and provides the initial term  $f_{11}$  of the sequence  $f_{xy}$ , similar to figure 3.2 and equation (3.18). As periodic boundary conditions identify the hatched pixels outside of the  $N \times N$  region with corresponding pixels on the other side of the region (cf. figure 2.2), all 4 corner pixels of the  $N \times N$  region are tagged after this initial step.

The following contributions of pixels of the first row are given in figure 3.11 and correspond to the contributions of minus sampling boundary conditions of figure 3.3. While in the case of minus sampling boundary conditions the hatched pixels did not contribute to anything but the first line, the hatched pixels have to be identified with the last line of pixels in the case of periodic boundary conditions and therefore do contribute to pixel configurations on the other side of the  $N \times N$  region. Thus, all hatched pixels need to keep their tags to account for their contributions to later configurations.

The final contribution of the first line of pixels is depicted in figure 3.12. All pixels in the red frame are already tagged — the left ones by the preceding step of processing pixels in the first line and the right ones via contributions



- light gray**  $\mapsto$  **brown** an unprocessed pixel is tagged, i.e. all summands of the operator are multiplied by  $b_{xy}$  or  $w_{xy}$ , depending on the configuration the summand describes
- brown**  $\mapsto$  **brown** previously tagged pixels may contain contributions in conflict with the desired configuration; these conflicting terms have to be removed by multiplying the summands of the operator by  $\hat{0}(b_{xy})$  or  $\hat{0}(w_{xy})$ , depending on the pixel state which does not contribute to the configuration described by the summand
- brown**  $\mapsto$  **dark gray** same as **brown**  $\mapsto$  **brown**, but there are no further conflicts, so operator can be multiplied by  $\hat{1}(b_{xy})\hat{1}(w_{xy})$  from the left to remove all remaining tags of this pixel
- light gray**  $\mapsto$  **dark gray** this pixel does not contribute to any other region, so there is no need for tagging or special treatment

Table 3.1: Mapping of color transitions to operator formulas

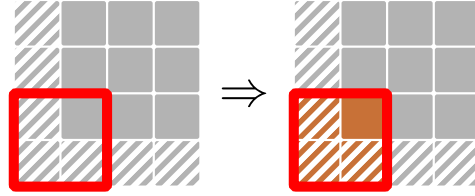


Figure 3.10: Schematic view of the contribution of the first pixel in the first row of pixels to the generating function of the number of equivalent tensor configurations using periodic boundary conditions. Hatched pixels lie on the outside of the  $N \times N$  pixels region and are used to determine the contribution of boundary pixels. The left hand side shows the state of the pixels before processing the red region, the right hand side afterwards. The color code is the same as in figure 3.1 and table 3.1 summarizes how to map the color transitions to operator formulas.

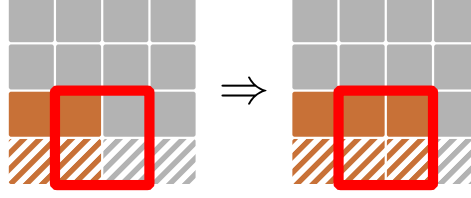


Figure 3.11: Schematic view of the contribution of a pixel in the first row of pixels to the generating function of the number of equivalent tensor configurations using periodic boundary conditions. Hatched pixels lie on the outside of the  $N \times N$  pixels region and are used to determine the contribution of boundary pixels. The left hand side shows the state of the pixels before processing the red region, the right hand side afterwards. The color code is the same as in figure 3.1 and table 3.1 summarizes how to map the color transitions to operator formulas.

from the first pixel of the row and the periodic boundary contributions. In contrast to figure 3.4 the processing of pixels stops one step earlier in figure 3.12, but it still covers all configurations, as shifting the red frame one pixel to the right would jump back to the contribution of the very first pixel via the periodic boundary conditions.

The second and all the following rows can be opened by the method depicted in figure 3.13. All rows opened by this operator can be processed by the interior pixel operator of equation (3.17) as described in section 3.3.1. The last contribution of every row is then given by figure 3.14.

When the last row is reached, all remaining pixels are already tagged due to their contributions to the configurations in the first line via the periodic boundary condition. How to open the last row of pixels, how to process its pixels and how to finalize it is shown in figures 3.15, 3.16 and 3.17 respectively.

Following similar approaches as shown in section 3.3.2 for minus sampling boundary conditions, or in this section for periodic boundary conditions, allows to cover a broad spectrum of boundary conditions when calculating distributions of Minkowski tensors. Additionally, the ideas shown here for periodic boundary conditions may be used to parallelize algorithms to compute

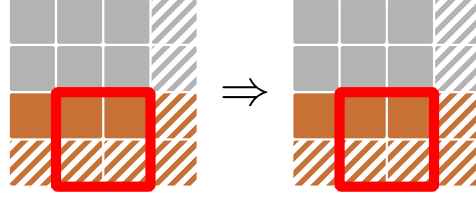


Figure 3.12: Schematic view of the contribution of the last pixel in the first row of pixels to the generating function of the number of equivalent tensor configurations using periodic boundary conditions. Hatched pixels lie on the outside of the  $N \times N$  pixels region and are used to determine the contribution of boundary pixels. The left hand side shows the state of the pixels before processing the red region, the right hand side afterwards. The color code is the same as in figure 3.1 and table 3.1 summarizes how to map the color transitions to operator formulas.

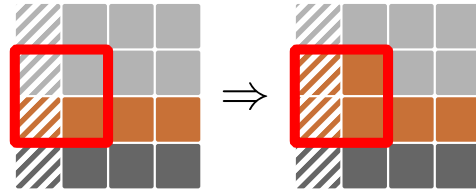


Figure 3.13: Schematic view of the contribution of the first pixel in a row of pixels to the generating function of the number of equivalent tensor configurations using periodic boundary conditions. Hatched pixels lie on the outside of the  $N \times N$  pixels region and are used to determine the contribution of boundary pixels. The left hand side shows the state of the pixels before processing the red region, the right hand side afterwards. The color code is the same as in figure 3.1 and table 3.1 summarizes how to map the color transitions to operator formulas.

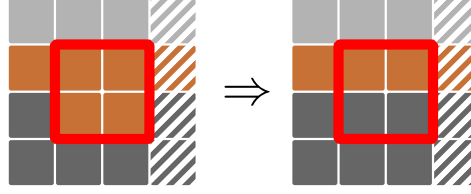


Figure 3.14: Schematic view of the contribution of the last pixel in a row of pixels to the generating function of the number of equivalent tensor configurations using periodic boundary conditions. Hatched pixels lie on the outside of the  $N \times N$  pixels region and are used to determine the contribution of boundary pixels. The left hand side shows the state of the pixels before processing the red region, the right hand side afterwards. The color code is the same as in figure 3.1 and table 3.1 summarizes how to map the color transitions to operator formulas.

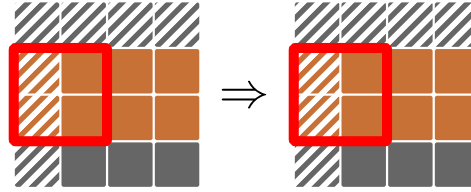


Figure 3.15: Schematic view of the contribution of the first pixel in the last row of pixels to the generating function of the number of equivalent tensor configurations using periodic boundary conditions. Hatched pixels lie on the outside of the  $N \times N$  pixels region and are used to determine the contribution of boundary pixels. The left hand side shows the state of the pixels before processing the red region, the right hand side afterwards. The color code is the same as in figure 3.1 and table 3.1 summarizes how to map the color transitions to operator formulas.

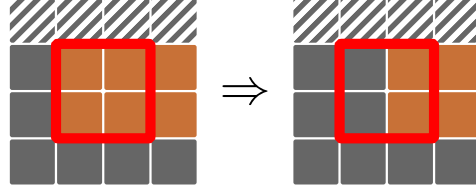


Figure 3.16: Schematic view of the contribution of a pixel in the last row of pixels to the generating function of the number of equivalent tensor configurations using periodic boundary conditions. Hatched pixels lie on the outside of the  $N \times N$  pixels region and are used to determine the contribution of boundary pixels. The left hand side shows the state of the pixels before processing the red region, the right hand side afterwards. The color code is the same as in figure 3.1 and table 3.1 summarizes how to map the color transitions to operator formulas.

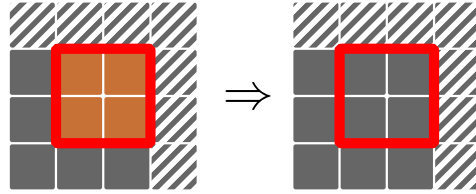


Figure 3.17: Schematic view of the contribution of the last pixel in the last row of pixels to the generating function of the number of equivalent tensor configurations using periodic boundary conditions. Hatched pixels lie on the outside of the  $N \times N$  pixels region and are used to determine the contribution of boundary pixels. The left hand side shows the state of the pixels before processing the red region, the right hand side afterwards. The color code is the same as in figure 3.1 and table 3.1 summarizes how to map the color transitions to operator formulas.

these distributions. Periodic boundary conditions may be seen as a change of topology, i.e. going from a lattice in euclidean space to a lattice on a torus. There is no boundary for the lattice on the torus, thus, the shown procedure provides a method to start somewhere in the middle of the lattice and a way to merge with such starting rows from the other side. This idea can be adopted for any set of boundary conditions on the euclidean lattice to divide the  $N \times N$  region in  $k$  strips of  $N \times N/k$  pixels. Each strip may be processed independently and the strips can be merged to provide the final generating function.

### 3.3.3 Construction of the Probability Distribution

While the construction of the generating function  $f(\{q_{ij}\})$  is by far more complex for a general combination of Minkowski tensors than for Area tensors (cf. section 3.2), the final steps to the common probability distribution of an arbitrary combination of Minkowski tensors are almost identical to the steps explained in section 3.2. Let  $\mathcal{T}$  be the set of Minkowski tensors to be considered and  $\mathcal{N}_{\mathcal{T}}(\{t_{ij}\}; N)$  the number of  $N \times N$  b/w images which lead to the same values for all the tensor components  $\{t_{ij}\}$  of tensors  $\mathbf{T}_j \in \mathcal{T}$ , where  $i$  enumerates the different components of the  $j^{\text{th}}$  tensor and  $j$  the tensors in  $\mathcal{T}$ . The partition function  $\mathcal{N}_{\mathcal{T}}(\{t_{ij}\}; N)$  is implicitly given via the constructed generating function

$$f(\{q_{ij}\}) = \sum_{t_{11}=-\infty}^{\infty} \cdots \sum_{t_{kl}=-\infty}^{\infty} \mathcal{N}_{\mathcal{T}}(\{t_{ij}\}; N) q_{11}^{t_{11}} \cdots q_{kl}^{t_{kl}}, \quad (3.33)$$

where  $k$  and  $l$  represent the indices of the last tensor component in  $\{t_{ij}\}$  and the ellipses have to be replaced by corresponding items for all elements in  $\{t_{ij}\}$ . The final probability  $\mathcal{P}_{\mathcal{T}}(\{t_{ij}\})$  to find a given combination of tensor values is then given by

$$\mathcal{P}_{\mathcal{T}}(\{t_{ij}\}) = \sum_{h=0}^M \mathcal{N}_{\mathcal{T}}(\{t_{ij}\}, h; N) p^h (1-p)^{M-h}, \quad (3.34)$$

where  $p$  is the probability of a pixel to be above the threshold and therefore black. The need for the new variables  $M$  and  $h$  arises from boundary effects and their motivation and exact definition is discussed in detail below. Despite this lack of information the interpretation of the formula is straightforward:

$h$  gives the number of pixels above threshold and  $M - h$  the number of pixels below threshold. Thus,  $p^h (1 - p)^{M-h}$  is the probability to find one configuration with the given amount of pixels above threshold. Multiplying this probability with the number of equivalent configurations  $\mathcal{N}_{\mathcal{T}}(\{t_{ij}\}, h; N)$  gives the probability to find the given set of tensor values  $\{t_{ij}\}$  in configurations with  $h$  pixels above threshold. Summing over all possible  $h$  finally gives the probability to find the given set of tensors in any pixel configuration.

In section 3.2 you could use  $h = A$  and  $M = N^2$ , i.e. the number of pixels above the threshold corresponded to the number of black pixels in the  $N \times N$  region. This is no longer true for minus sampling boundary conditions. With minus sampling boundary conditions pixels on the outside of the  $N \times N$  region do not contribute to  $A$ , but do influence the probability of a given configuration as the probability of an exterior pixel to be above the threshold is the same as for interior pixels, i.e. black boundary pixels contribute to  $h$  and you need to take all the boundary pixels into account, thus,  $M = (1 + N + 1)^2$ . On the other hand  $h$  is not always the extended area  $\bar{A}$ , i.e. the area including all boundary pixels. If you choose to set the exterior pixels to fixed colors, e.g. setting all exterior pixels to black or to a check pattern, then the exterior pixels contribute to  $\bar{A}$  but not to the probability of a given configuration and therefore not to  $h$ . So, for boundary conditions with fixed exterior colors or for periodic boundary conditions you really find  $h = A$  and  $M = N^2$ . Therefore, you need to determine  $M$ , the number of variable pixels, according to your chosen boundary conditions and you need to track  $h$ , the number of variable pixels above threshold, in addition to all the tensor components  $\{t_{ij}\}$  when constructing the generating function and the corresponding partition function. If you end up finding  $h = A$ , equation (3.34) still holds, as then  $\mathcal{N}_{\mathcal{T}}(\{t_{ij}\}, h; N) = 0 \forall h \neq A$  — they are tracking the same value after all and there can be no configuration with different contributions to the two variables — and thus the sum collapses to the  $h = A$  term if  $A \in \mathcal{T}$ .

So, in general it is not sufficient to determine the partition function  $\mathcal{N}_{\mathcal{T}}(\{t_{ij}\}; N)$  for a given set of tensors  $\mathcal{T}$ , you also need to keep track of  $h$ , the number of variable pixels above the threshold. After constructing  $\mathcal{N}_{\mathcal{T}}(\{t_{ij}\}, h; N)$  via the generating function formalism introduced before, the final probability distribution of an arbitrary set of Minkowski tensors  $\mathcal{T}$  is simply given by equation (3.34).

### 3.4 Comparison to Brute Force Approach

It seems plausible to assume that the algorithm using generating functions is more efficient than a plain brute force approach. Generating functions can be computed successively by exploiting the additivity of the Minkowski tensors, which should reduce the required overhead significantly. Nevertheless, up to now these are just plausible assumptions and there is no hard evidence for improved performance. In fact it is quite hard to prove the efficiency of the given algorithm in general, as the efficiency depends on the set of chosen Minkowski tensors and may even show a worst case performance similar to the brute force approach. After discussing this in the context of some general complexity studies, it is shown that the algorithm is usually very efficient. After giving some examples to show the possible performance gain in an actual implementation of the algorithm, some technical details and approaches to improve the performance are discussed.

#### 3.4.1 Complexity Analysis

The coefficients of the final probability distribution and therefore also the coefficients and exponents of the generating function are of the same order of magnitude as the number of possible configurations, i.e.  $\mathcal{O}(2^{N^2})$ . Thus, starting from  $N = 6$  the values do not fit in 32 bit integers and after  $N = 8$  they are too large for 64 bit integers. To prevent integer overflows and similar complications you should use arbitrary precision libraries to hold your values. However, this may lead to performance penalties, as you can no longer assume additions and multiplications to be constant time operations. The following discussion ignores this, as it affects both, the brute force and the generating function algorithm, and the discussion merely focuses on differences regarding the complexity of the algorithms.

The brute force approach has to process all of the  $2^{N^2}$  configurations and for each configuration it has to calculate the values of all the required tensors. As the tensors are additive, the calculation can be done by processing  $2 \times 2$  regions (cf. section 2.2.3). Assuming constant time for each  $2 \times 2$  region<sup>2</sup> each configuration needs  $\mathcal{O}(N^2)$  operations to determine the values of all

---

<sup>2</sup>This can be achieved e.g. by implementing the evaluation as lookup in a table of all possible  $2 \times 2$  configurations at the origin and using equation (2.25) to shift the coordinates to the appropriate target region.



tensors. Combined with the number of configurations to process this leads to  $\mathcal{O}(N^2 2^{N^2})$  complexity of the brute force algorithm.

The generating function algorithm processes pixel by pixel successively. For every step all 16 configurations of the current  $2 \times 2$  region have to be computed, which can be done in constant time by precomputing the configurations at the origin and using equation (2.25) to shift the values to the current coordinates. So, computing the operator  $\hat{O}_{xy}$  of equation (3.17) is a constant time operation for every pixel. Applying  $\hat{O}_{xy}$  as described in equation (3.13) is linear in the number of summands of  $f_{(x-1)y}$ . This can be seen by decomposing the operator  $\hat{O}_{xy}$  into its 16 summands  $\hat{X}_i$ , applying each  $\hat{X}_i$  individually and summing the 16 resulting polynomials. The applications of one of these  $\hat{X}_i$  is linear, as the result can be computed by looping over the summands in  $f_{(x-1)y}$ , dropping the current summand if  $\hat{X}_i$  contains a corresponding  $\hat{0}$  operator, adding appropriate factors of  $q_{ij}$ ,  $b_{xy}$  and  $w_{xy}$  and finally removing tag variables if there are corresponding  $\hat{1}$  operators in  $\hat{X}_i$ . All the required sums can also be computed in linear time, leading to an overall complexity of  $\mathcal{O}(k)$  of applying  $\hat{O}_{xy}$ , where  $k$  is the number of summands in  $f_{(x-1)y}$ . Unfortunately it is not generally possible to predict  $k$  for every step of the algorithm, as it depends on the set of chosen tensors  $\mathcal{T}$ . If  $\mathcal{T} = \{A\}$ ,  $k$  is the number of pixels processed so far, as every pixel adds exactly 1 new configuration ( $A = k + 1$ ) to all the configurations known so far ( $A \in \{0, \dots, k\}$ ); using  $\sum_{k=0}^{N^2} k = \frac{N^2(N^2+1)}{2}$  this leads to an overall complexity of the algorithm of  $\mathcal{O}(N^4)$ . If on the other hand  $\mathcal{T}$  contains enough tensors to uniquely identify every possible configuration, you find  $k = 2^i$ , where  $i$  is the number of pixels processed so far. This is due to the fact that every pixel doubles the number of possible configurations (the pixel may be black or white) and all configurations lead to distinct terms in the polynomial, as the tensors are chosen to be different for every configuration, which prevents simplifications of terms. Using  $\sum_{i=0}^{N^2} 2^i = 2^{N^2+1} - 1$  this leads to an overall complexity of  $\mathcal{O}(2^{N^2})$ , which is only slightly better than the brute force approach.

While this proves that the generating function algorithm is asymptotically better than the brute force approach, independent of the chosen tensors  $\mathcal{T}$ , it does not tell much about its real performance and usability for small  $N$ . The next section clarifies this by showing some benchmark results of both algorithms.

### 3.4.2 Benchmark Results

The algorithms discussed in this section were all implemented using Java. While there still exist lots of prejudices against Java as a tool for scientific computing, this language was chosen over alternatives like C++ for several reasons. First of all the common “Java is slow” claim is a relic from the early days of the language, when all the code was interpreted and the performance could not compete with compiled languages like C++. Nowadays Java virtual machines (JVM) use just in time (JIT) compilation to run native machine code instead of interpreted byte code. Modern JIT compilers are very efficient and can exploit optimization techniques not accessible by conventional ahead of time compilers, like dynamic recompilation<sup>3</sup> or whole program optimization.<sup>4</sup> Using a recent JVM with powerful JIT compiler, Java performance is on par with (or even better than) C++ performance (Wikipedia, 2012) and can even compete with Fortran on number crunching tasks (Amedro et al., 2008). Another big point for Java are the superior tools available for developers, like IDEs with code completion, simple navigation (jump to definition, show all references, ...), support for refactoring, integration of graphical debuggers and unit testing frameworks. The tools shipped with the language, i.e. the standard library, provide lots of useful building blocks not available in the C++ standard libraries, like arbitrary precision arithmetic, support for thread pools and concurrent collections. Finally, the most common source for program crashes in C++, the infamous segfault, just does not exist in Java. All in all Java allows more rapid and flexible development while keeping the performance on the highest levels.

While there are many good reasons to use Java for implementing scientific algorithms, the dynamic nature of the JVM and its JIT compiler pose some problems for benchmarks and make it harder to quantify the efficiency. The pitfalls of benchmarking Java code are discussed in Boyer (2008) and the author provides an open source library to benchmark Java code, which avoids these problems and gives solid feedback on the performance of the implementation under test. This benchmarking framework was used to profile the different

---

<sup>3</sup>Based on profiling data of the running application parts of the code can be recompiled to optimize the program for the current requirements.

<sup>4</sup>The JIT compiler knows all the used code, while ahead of time compilers have to assume arbitrary use cases for library code. This allows more specific optimization than generic ahead of time compilation.

algorithms of calculating Minkowski tensor distributions.

The benchmarks were run on the TinyFat cluster (RRZE, 2011) of the RRZE, the high performance computing center of the university Erlangen-Nuremberg. Each node of this cluster provides 16 cores of 2.3 GHz and 128 GB of main memory. The JVM used on the cluster is the official Java SE 7u2 release of Oracles Java Runtime Environment.

Three different algorithms were benchmarked to study their different performance characteristics. The brute force algorithm — evaluating all possible b/w configurations and counting the configurations with equal Minkowski tensor values — is denoted by  $\text{alg}_{\text{BF}}$ . The second version is the algorithm discussed in section 3.2 for calculating the distributions of area tensors, which requires no additional bookkeeping using tag variables like  $b_{xy}$  or  $w_{xy}$ . This variant is denoted by  $\text{alg}_{\text{AT}}$ . The general algorithm described in section 3.3 to compute distributions of arbitrary sets of Minkowski tensors using tag variables is denoted by  $\text{alg}_{\text{GA}}$ .

Figure 3.18 shows the results of the benchmark runs. The brute force algorithm  $\text{alg}_{\text{BF}}$  was only considered for one set of tensors, namely  $\{A, P, \chi\}$ , as the run-time is independent of the chosen tensors — it always processes all configurations. While  $\text{alg}_{\text{BF}}$  is almost as fast as the other algorithms for  $N = 3$ , it is the only one with a super-exponential increase of the run-time<sup>5</sup> and thus it is always the slowest and its rapidly rising run-time prevents its usage even for intermediate region sizes.<sup>6</sup>

The area tensor algorithm is shown for two configurations:  $\text{alg}_{\text{AT}}(A)$  gives the performance for computing the distribution of the area only and  $\text{alg}_{\text{AT}}(A, \vec{A}_1, \mathbf{A}_2)$  the performance of computing the common distribution of the area functional, its center of mass vector and the area tensor of rank 2. The area tensors up to rank 2 contain enough information to uniquely identify almost all configurations of small regions. Thus,  $\text{alg}_{\text{AT}}(A, \vec{A}_1, \mathbf{A}_2)$  should reflect the worst case performance of  $\text{alg}_{\text{AT}}$  (cf. section 3.4.1), while the area

---

<sup>5</sup>Exponential increase of the run-time corresponds to a linear rise in the given logarithmic plot. Thus, positive curvature in the plot corresponds to super-exponential increase and negative curvature to sub-exponential increase.

<sup>6</sup>Even the value for  $N = 6$  had to be extrapolated from runs which processed about 8% of the required configurations, as the predicted run-time is already  $\mathcal{O}(1 \text{ month})$  and it would not have been feasible to block cluster nodes for such benchmark runs. All other data points in figure 3.18 are direct measurements using the benchmark framework of Boyer (2008).

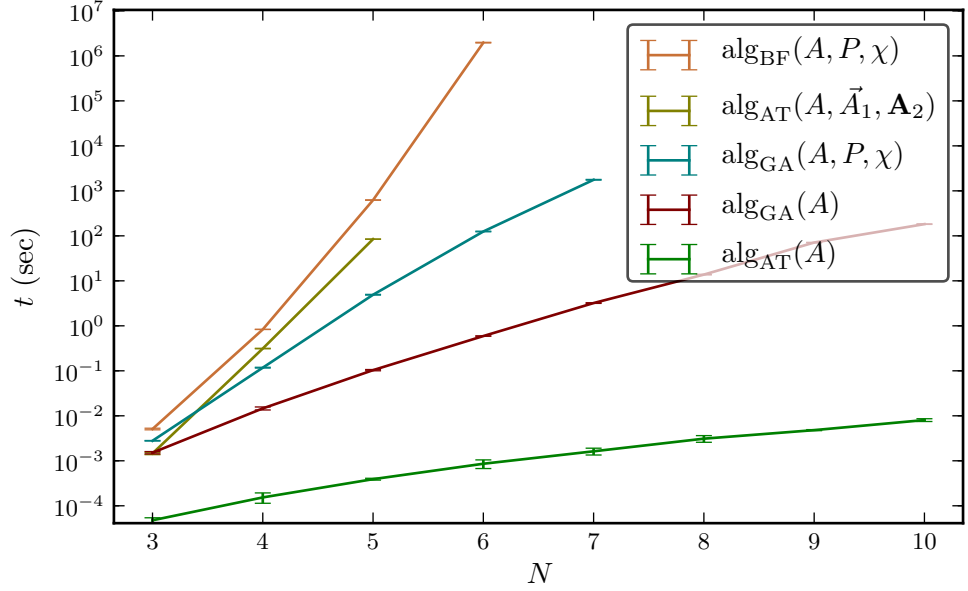


Figure 3.18: Run-time of the different algorithms in seconds over the size of the  $N \times N$  pixel region whose distribution is to be computed. The legend gives the acronym of the used algorithm and the set of Minkowski tensors, for which the common distribution was computed.

functional on its own contains the least amount of information, i.e.  $\text{alg}_{\text{AT}}(A)$  provides a best case performance estimate. Unfortunately the performance of  $\text{alg}_{\text{AT}}(A, \vec{A}_1, \mathbf{A}_2)$  could only be tested up to  $N = 5$ , as the memory requirements of the algorithm outgrew the available hardware at  $N = 6$  (cf. the discussion of figure 3.19 for details on memory usage). Based on the 3 given data points it is impossible to reliably infer super- or sub-exponential growth of the run-time, but given the initial points and the asymptotic analysis in section 3.4.1,  $\text{alg}_{\text{AT}}$  seems to be more efficient than  $\text{alg}_{\text{BF}}$  for all  $N$ .

The additional bookkeeping required to compute distributions for arbitrary sets of tensors leads to larger intermediate polynomials which need to be multiplied, which in turn leads to increased run-times. The comparison of  $\text{alg}_{\text{AT}}(A)$  and  $\text{alg}_{\text{GA}}(A)$  shows that this overhead is significant. Nevertheless,

the additional tag variables may prevent simplification of terms, but they do not introduce genuinely new terms, i.e. if the tensor components suffice to distinguish all possible configurations, the intermediate polynomials are of maximal length and the overhead of tag variables negligible. Thus, the order of magnitude of worst case performance of  $\text{alg}_{\text{GA}}$  should correspond to  $\text{alg}_{\text{AT}}(A, \vec{A}_1, \mathbf{A}_2)$ . Last but not least  $\text{alg}_{\text{GA}}(A, P, \chi)$  shows the performance of the algorithm for area, perimeter and Euler characteristic, a set of functionals that is of special interest in later sections. There are no values for  $\text{alg}_{\text{GA}}(A, P, \chi)$  after  $N = 7$ , as further computations would have required more memory than available on the cluster used for benchmarking.

In general it is quite hard to measure the memory requirements of a garbage collected language like Java. Increasing the memory pressure forces the garbage collector (GC) to look harder for reclaimable memory, which might reduce the memory usage significantly but might also slow down the program due to significant GC load. Without reliable means to free unneeded memory it is impossible to distinguish between memory used by live objects and “leaked” memory which got not collected yet. The problem gets even more complicated by the fact that the current implementations use lots of caches in their computations. These caches were introduced when it became apparent that the algorithms in general need lots of RAM. As the computations are done using arbitrary precision arithmetic, all the involved numbers are complex objects. Lots of these objects are equal, i.e. represent the same numeric value, but are not the same, i.e. occupy individual pieces of memory. By using only one instance of every number and references to this instance where ever the number is needed, the memory usage could be reduced significantly. These unique instances need to be managed in a central pool, which prevents or at least delays garbage collection of these objects.<sup>7</sup> Instead of accounting for all these different effects, figure 3.19 shows some upper limits on the required memory for the different algorithms. These upper limits may include uncollected memory and unneeded cache values, but provide an estimate of the order of magnitude of needed resources. The shown data was extracted from the memory usage of the benchmarking runs shown in figure 3.18.

All upper limits start at  $\mathcal{O}(0.5 \text{ GB})$ , as this is the initial heap size of the

---

<sup>7</sup>Internally so called SoftReferences are used to build these caches. Soft references may be collected by the GC if the system runs low on memory, but are kept as long as possible. This way the size of the caches is dynamically tuned by the GC to fit the size of the available memory.

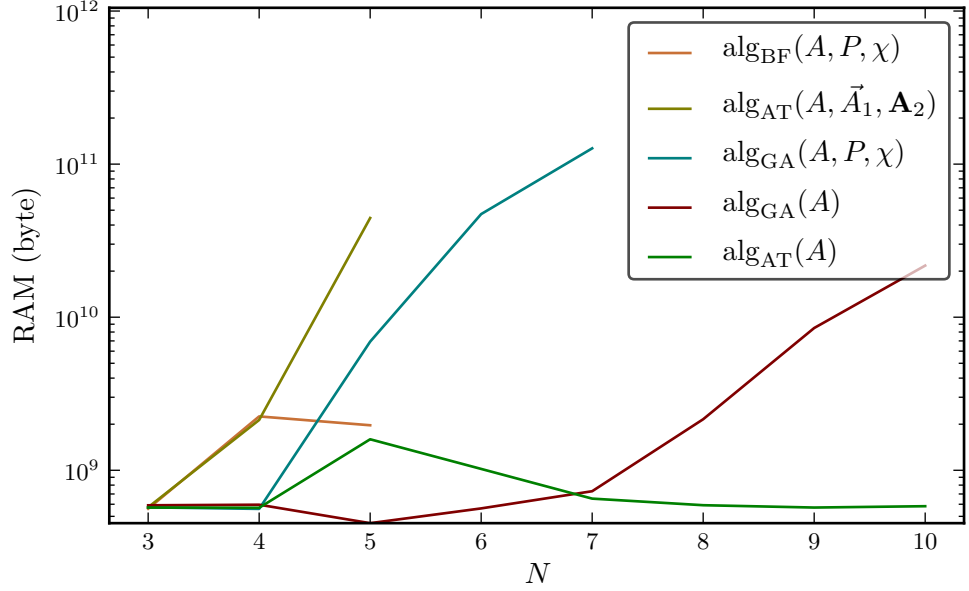


Figure 3.19: Upper limit of the memory usage of the different algorithms in bytes over the size of the  $N \times N$  pixel region whose distribution is to be computed. The legend gives the acronym of the used algorithm and the set of Minkowski tensors, for which the common distribution was computed.

JVM on the used hardware and there are effectively no GC runs as long as all objects fit in this initial memory.  $\text{alg}_{\text{BF}}(A, P, \chi)$  is only shown up to  $N = 5$ , as the run-time for larger  $N$  is too big to get a direct measurement of the final resource usage. The lines for  $\text{alg}_{\text{AT}}(A, \vec{A}_1, \mathbf{A}_2)$  and  $\text{alg}_{\text{GA}}(A, P, \chi)$  are only shown up to  $N = 5$  and  $N = 7$  respectively, as the algorithm needs more than the available 120 GB to compute distributions for larger  $N$ . Especially the comparison of  $\text{alg}_{\text{AT}}(A, \vec{A}_1, \mathbf{A}_2)$  and  $\text{alg}_{\text{AT}}(A)$  shows that the memory usage depends strongly on the set of chosen tensors. The computed distribution has to map every possible tensor configuration to the corresponding probability and thus the size to store the distribution increases with the number of possible tensor configurations.

So, while  $\text{alg}_{\text{BF}}$  was limited by CPU power and program run-time, the new

algorithms solve this issue and make distributions for larger  $N$  accessible. Unfortunately the new algorithms need quite some memory for internal bookkeeping and hit the next hardware limit by requiring excessive amounts of RAM. It might be possible to push the limits further by offloading the in memory data structures to hard disk backed database engines. Some initial tries with relational database systems were performed, but the performance of the resulting implementations dropped below bearable values. Using NoSQL alternatives like Cassandra (The Apache Software Foundation, 2008) might provide better throughput due to their reduced overhead, but so far this approach was not investigated any further. In addition to putting the data structures in databases instead of plain Java maps it might be beneficial to use solid state disks instead of hard disk drives as storage back end of the data base, which should increase the throughput significantly. Using a specialized storage system to offload the in memory data structures to mass storage could pave the way to larger  $N$  while keeping the run-time in reasonable boundaries, but this approach needs further studies to validate these claims.

As approaches to reduce the memory footprint might result in significantly longer run-times (disk I/O is of  $\mathcal{O}(1000)$  times slower than main memory I/O), it is still interesting to look for ways to reduce the run-time beyond the current achievements. In fact, all of the discussed algorithms can be parallelized quite efficiently. As  $\text{alg}_{\text{BF}}$  processes all possible configurations individually, it is straightforward to parallelize this algorithm by processing disjoint subsets of the configurations in different threads and combining the individual thread results at the end. The way to parallelize  $\text{alg}_{\text{GA}}$  is somewhat more involved and was already discussed at the end of section 3.3.2: The  $N \times N$  region can be divided into strips which may be processed individually. The resulting strips contain overlapping tag variables and need to be merged by the methods described in section 3.3.2.

While these approaches help to exploit the full potential of modern multi-core CPUs, they also increase the memory requirements. Separating the algorithms into independent sub-tasks prevents the simplification of common terms and each sub-task has to manage its own copy of these common terms. So you have to find the right balance between memory and CPU requirements when selecting the optimal strategy to implement the different algorithms.

All in all it could be shown that  $\text{alg}_{\text{GA}}$  is always more efficient than  $\text{alg}_{\text{BF}}$ . Unfortunately it is no magic cure which immediately provides distributions for arbitrary tensors and region sizes, but it makes new distributions accessible

which were not computable before and might provide a powerful tool to compute distributions on future hardware.



## 4 Minkowski Maps

The previous chapters show how to compute Minkowski tensors and how to derive their probability distributions given a homogeneous and isotropic background model. This chapter shows how to combine both techniques to quantify the significance of structural deviations in measured images compared to the given background model expectations.

### 4.1 Quantification of Structure Deviations

As the distributions of Minkowski tensors for the background model are known, it is possible to test the null hypothesis  $H_0$ :

The measured image shows only background structures.

The significance level  $\alpha$  at which this hypothesis can be rejected may be used as indication on how much a measured image deviates from the typical structure of a background image.

In general, null hypothesis tests are based on the choice of a critical region  $\omega$  in the sample space  $\Omega$ , i.e. a region in the space of possible outcomes where the hypothesis shall be rejected. The significance of a hypothesis test is then given by

$$\alpha = \mathcal{P}_{H_0}(\omega), \quad (4.1)$$

i.e. by the probability that  $H_0$  is rejected while being true (error of the first kind). Let  $H_{\vec{t}}$  be an alternative hypothesis assigning the probability (density)  $\mathcal{P}_{H_{\vec{t}}}(x)$  to every outcome  $x \in \Omega$ , where  $\vec{t}$  is the vector of parameters of this probability mass/density function. Let further  $\mathcal{P}_{H_0}(x)$  be the probability mass/density function associated with  $H_0$ . An efficient critical region for testing  $H_0$  on the significance level  $\alpha$  is a region which leads to a small probability of an error of the second kind, i.e. of accepting  $H_0$  while  $H_{\vec{t}}$  is true. Consider e.g. the following choice of a critical region for rejecting  $H_0$  in

favor of  $H_{\vec{t}}$ :

$$\omega = \left\{ x \in \Omega \mid \frac{\mathcal{P}_{H_0}(x)}{\sup_{\vec{t}} \mathcal{P}_{H_{\vec{t}}}(x)} \leq k(\alpha) \right\}, \quad (4.2)$$

i.e. the set of all points  $x \in \Omega$  for which the shown likelihood ratio is smaller than a given constant  $k$ , where  $k$  has to be chosen such that  $\alpha = \mathcal{P}_{H_0}(\omega)$  for the desired significance level  $\alpha$ . Here likelihood ratio means the ratio of the probability (density) of  $x$  assuming  $H_0$  is true and the supremum of all alternative probabilities/probability densities of  $x$ . Neyman and Pearson (1933) have shown that while critical regions chosen like this always provide efficient criteria to reject hypotheses, they even provide the most efficient criteria for a large class of tests. So, while these critical regions always lead to small probabilities of errors of the second kind they often even lead to the smallest possible probability of an error of the second kind for a given significance level  $\alpha$ .

If there are no further constraints on  $H_{\vec{t}}$ , it also includes the hypothesis where  $x_0 \in \Omega$  is the certain event and all other outcomes are impossible. As  $H_{\vec{t}}$  comprises such hypotheses for all outcomes  $x \in \Omega$ ,  $\sup_{\vec{t}} \mathcal{P}_{H_{\vec{t}}}(x)$  always selects the parameters  $\vec{t}$  for which  $x$  is the certain event and thus yields a constant independent of  $x$ . After absorbing this constant in the constant  $k$  the efficient critical region for testing  $H_0$  without any constraints on possible alternative hypotheses is given by

$$\omega = \{x \in \Omega \mid \mathcal{P}_{H_0}(x) \leq k(\alpha)\}. \quad (4.3)$$

So, if  $x_0$  is the measured outcome of an experiment, then

$$\omega(x_0) = \{x \in \Omega \mid \mathcal{P}_{H_0}(x) \leq \mathcal{P}_{H_0}(x_0)\} \quad (4.4)$$

is the smallest of these critical regions containing  $x_0$  and thus the highest possible significance level at which  $H_0$  may be rejected is given by

$$\alpha(x_0) = \mathcal{P}_{H_0}(\omega(x_0)). \quad (4.5)$$

In this sense  $\alpha(x_0)$  is a measure of how significantly  $x_0$  deviates from the null hypothesis  $H_0$ .

For  $\Omega = \mathbb{R}$  and  $\mathcal{P}_{H_0}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$  this general approach recovers the well known significance  $\alpha(n) = 1 - \text{erf}(n/\sqrt{2})$  of a  $n\sigma$  deviation from the

expectation value of a normal distribution. While this concept of deviations from the expectation value is hard to generalize to asymmetric or multivariate probability distributions, equation (4.5) can readily be applied to arbitrary probability distributions. In general  $\alpha(x)$  describes the probability to measure any outcome which is at most as probable as  $x$  assuming  $H_0$  is true and thus quantifies the compatibility of the measured value with the null hypothesis  $H_0$ . The compatibility of the most probable outcome predicted by  $H_0$  is always 1, no matter what probability mass/density function describes  $H_0$ , and the compatibility of the least probable outcome predicted by  $H_0$  depends on the number of possible realizations of this least probability. The uniform distribution over the interval  $[a; b]$  e.g. yields a compatibility of 1 for all values within the given interval and 0 beyond the interval, in agreement with the intuitive expectation.

Using this framework it is straightforward to quantify the compatibility of a measured gamma-ray counts map with the structure of a background measurement. Let  $\mathcal{T}$  be the set of Minkowski tensors to be considered and  $\mathcal{P}_{\mathcal{T}}(\{t_{ij}\}; N)$  the probability to measure the tensor components  $\{t_{ij}\}$  in a  $N \times N$  b/w image which shows only background, as derived in section 3.3.3, where  $i$  enumerates the different components of the  $j^{\text{th}}$  tensor and  $j$  the tensors in  $\mathcal{T}$ . The compatibility of measured tensor components  $\{t_{ij}\}$  with the null hypothesis of a pure background measurement is then given by

$$\alpha_{\mathcal{T}}(\{t_{ij}\}; N) = \sum_{\{t_{kl}\} \in \omega_{\mathcal{T}}(\{t_{ij}\}; N)} \mathcal{P}_{\mathcal{T}}(\{t_{kl}\}; N), \quad (4.6)$$

where  $\omega_{\mathcal{T}}(\{t_{ij}\}; N) = \{\{t_{kl}\} | \mathcal{P}_{\mathcal{T}}(\{t_{kl}\}; N) \leq \mathcal{P}_{\mathcal{T}}(\{t_{ij}\}; N)\}$ , i.e. the set of all tensor configurations which are at most as probable in the background model as the measured tensor components.

The most interesting configurations are those with significant deviations from the structure of a background measurement or with small compatibility with the background model respectively. Hence, the most interesting values of  $\alpha_{\mathcal{T}}$  are tiny. Therefore, it is convenient to introduce the deviation strength

$$\mathcal{D}_{\mathcal{T}}(\{t_{ij}\}; N) = -\log_{10}(\alpha_{\mathcal{T}}(\{t_{ij}\}; N)). \quad (4.7)$$

Table 4.1 shows some values of significances in different units and especially the deviation strength corresponding to the commonly used  $5\sigma$  criterion.

While the previous paragraphs show how to determine the deviation strength of a single b/w pixel image, you need to quantify the deviation strength of

$\sigma$	$\mathcal{D}$	$1 - \alpha$	$\alpha$
0	0.0000000000000000	0.0000000000000000	1.0000000000000000
1	0.498515545827989	0.6826894921370859	0.3173105078629141
2	1.341986084476956	0.9544997361036416	0.0455002638963584
3	2.568669040265385	0.9973002039367398	0.0026997960632602
4	4.198304911892497	0.9999366575163338	$6.3342483666 \cdot 10^{-5}$
5	6.241615676743596	0.9999994266968563	$5.7330314374 \cdot 10^{-7}$
6	8.704834331867469	0.999999980268247	$1.9731752898 \cdot 10^{-9}$

Table 4.1: Significances in different units.  $\sigma$  denotes the number of standard deviations of a normally distributed random variable which corresponds to the deviation of the given observation from the null hypothesis.  $\mathcal{D}$  denotes the deviation strength as defined in equation (4.7).  $\alpha$  denotes the probability of an error of the first kind, i.e. the compatibility of the observed outcome with the null hypothesis.

a series of b/w images, in order to fully judge the structure of a gray scale image. So, if you create a series of b/w images from a gamma-ray counts map via thresholding (cf. section 2.1.4) the null hypothesis should not just describe some of these images, but needs to be compatible with all of them. Therefore, each of the different b/w images contributes a trial to the search for structure deviations and as the number of trials increases, the probability to find a significant random fluctuation increases. Suppose you look for measurements with a compatibility of less than  $\alpha$ , then the probability to find such a deviation in one image is  $\alpha$ . The probability to find no such deviation in  $t$  independent images is  $(1 - \alpha)^t$  and the probability to find at least one such deviation in the image set is  $1 - (1 - \alpha)^t$ . Thus, the significance after  $t$  independent trials is linked to the pre-trial significance  $\alpha$  via

$$\begin{aligned}
\alpha_t &= 1 - (1 - \alpha)^t \\
&= 1 - \sum_{i=0}^t \binom{t}{i} (-\alpha)^i \\
&= t\alpha + \mathcal{O}(\alpha^2).
\end{aligned} \tag{4.8}$$

So, for  $\alpha \ll 1$  the influence of  $t$  independent trials may be approximated by the so called trial factor, i.e. by multiplying  $\alpha$  with  $t$ . For the corresponding

deviation strength  $\mathcal{D}_t$  this results in an offset of  $\log_{10}(t)$  compared to the pre-trial deviation strength  $\mathcal{D}$ , i.e.  $\forall \mathcal{D} \gg 0 : \mathcal{D}_t \approx \mathcal{D} - \log_{10}(t)$ .

Obviously the different b/w images resulting from thresholding are not statistically independent, as they originate from the same intensity profile. Figuratively speaking, if the structure of a b/w image at a certain threshold can be interpolated from the structures of the b/w images of the next higher and lower thresholds, it does not contribute a separate trial to the search for deviations from the null hypothesis. Therefore, if  $t$  is set to the number of threshold steps, equation (4.8) will yield a conservative estimate of the post-trial significance.

A rough estimate of the systematic error on the deviation strength introduced by ignoring the influence of trial factors is given by  $2\log_{10}(N)$ . This estimate is based on the fact that the number of different b/w images after thresholding is equal to the number of different gray levels in the original gray scale image representing the gamma-ray counts map. There are at most  $N^2$  different gray levels in an image of  $N \times N$  pixels and thus at most  $N^2$  different trials may contribute to the search for structure deviations:

$$\begin{aligned} t &\leq N^2 \\ \mathcal{D}_t &\approx \mathcal{D} - \log_{10}(t) \\ &\geq \mathcal{D} - 2\log_{10}(N). \end{aligned} \tag{4.9}$$

Therefore,  $\mathcal{D} - 2\log_{10}(N)$  may be used to compute a conservative estimate of the post-trial deviation strength. Given that exact distributions could only be computed up to  $N \approx 7$  on the available hardware (cf. section 3.4.2) the error on the deviation strength introduced by ignoring trial factors is less than 1.7 and thus trial factors are ignored in the following sections. If a rigorous treatment of trial factors is required, the proper values can be determined via Monte Carlo simulations.

To summarize, for a series of threshold dependent b/w images the compatibility of each image to the null hypothesis of a pure background structure is computed. The least compatible image is selected to represent the compatibility of the given series, but this selection has to be accounted for via so called trial factors. The exact determination of trial factors needs detailed Monte Carlo simulations to determine their value for all needed combinations of image sizes  $N$ , Minkowski tensors  $\mathcal{T}$  and background levels  $\lambda$  (cf. equation (1.1)). However, as this work aims to present the general ideas of structure

analysis instead of providing high precision results, trial factors are ignored in the following for simplicity. This introduces a systematic uncertainty of  $\mathcal{O}(\log_{10}(N))$  into all shown results, which you need to be aware of when interpreting given numbers.

### 4.2 Local Structure Analysis

Instead of analyzing the entire gamma-ray counts map at once, it is often convenient to examine smaller regions. By using an  $N \times N$  pixels scan window to examine the local structure of all  $N \times N$  pixel regions of the counts map, it is possible to localize structure deviations and to probe their extension and shape. As Minkowski tensors provide a measure of extension and shape, it is in principle possible to infer the underlying shape of a potential gamma-ray source using the measured Minkowski tensors of the entire map. Nevertheless, these measured tensor components contain contributions from the superimposed background structure. Instead of finding ways to remove these contributions from the tensors of the entire map, it is easily possible to draw a map of local structure deviations from the expected background structure and to recover source morphologies from these Minkowski maps. Besides providing information on position and shape of potential gamma-ray sources, using smaller scan windows alleviates the problem that distributions for large  $N$  are hard to compute.

The following series of images depicts the steps required to create Minkowski maps from square pixelated intensity distributions. Figure 4.1 shows the initial intensity distribution. Of this initial distribution the highlighted  $5 \times 5$  pixel region in figure 4.2 is used to determine the local structure of the image in this region. For this local region the deviation strength of each b/w image is determined for the corresponding threshold  $\rho$ . The maximum deviation strength reflects the overall compatibility of the null hypothesis of a pure Poisson background intensity distribution with the local structure of the current region. Therefore, the maximum of the deviation strength over all thresholds is filled in the Minkowski map at the center of the  $5 \times 5$  region to depict the compatibility of the background hypothesis with the local structure around this pixel, as shown in figure 4.3. Performing this local analysis for all  $5 \times 5$  regions of the initial image leads to a Minkowski map as shown in figure 4.4. For a quantitatively exact analysis you would have to correct the

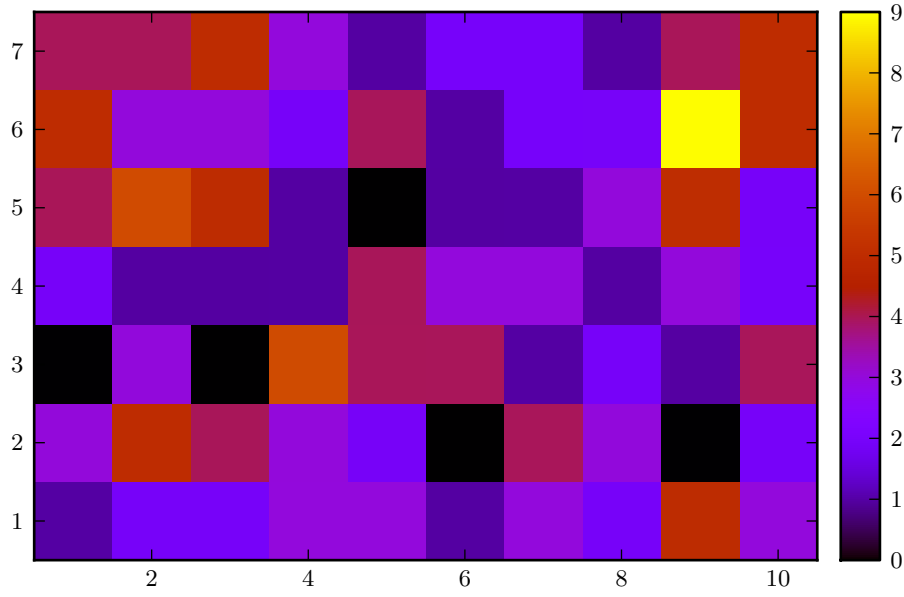


Figure 4.1: Homogeneous and isotropic Poisson distributed background map. Each pixel shows a random number of counts  $k$  sampled from the Poisson distribution  $\frac{\lambda^k}{k!} e^{-\lambda}$  with  $\lambda = 2.74$  and thus represents a possible background measurement in gamma-ray astronomy. The coordinate system just numbers the pixels of this simulated background measurement.

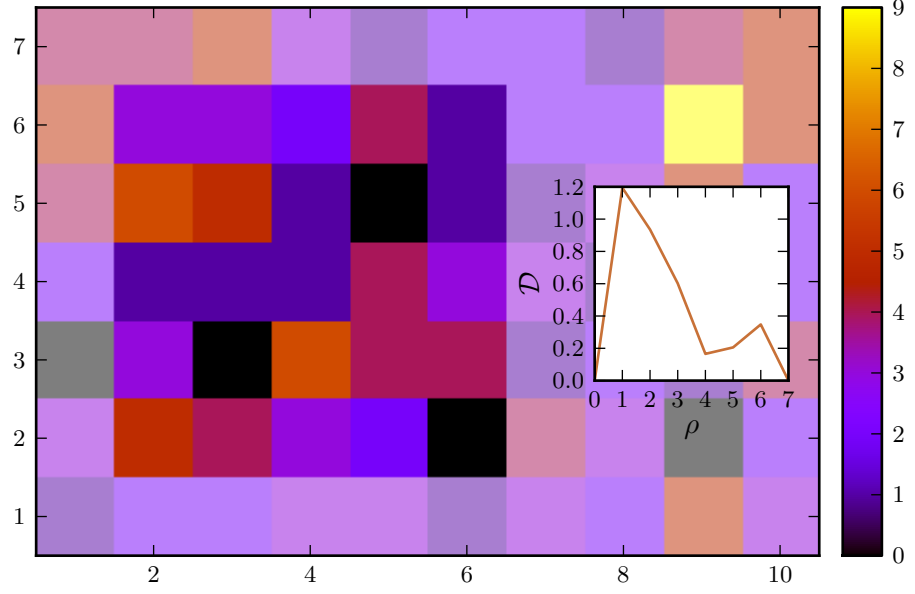


Figure 4.2: Analysis of the local structure of a  $5 \times 5$  pixels region of the background measurement of figure 4.1. The relevant region is highlighted by fading the other pixels to white. The inset shows the deviation strength of the highlighted region over the applied threshold for b/w conversion. The Minkowski tensors used to evaluate the structure are the area  $A$ , the perimeter  $P$  and the Euler characteristic  $\chi$ . The probability  $p$  of a pixel to be black is determined by the background model and the threshold via  $p = \sum_{k=\lceil \rho \rceil}^{\infty} \frac{\lambda^k}{k!} e^{-\lambda}$ .



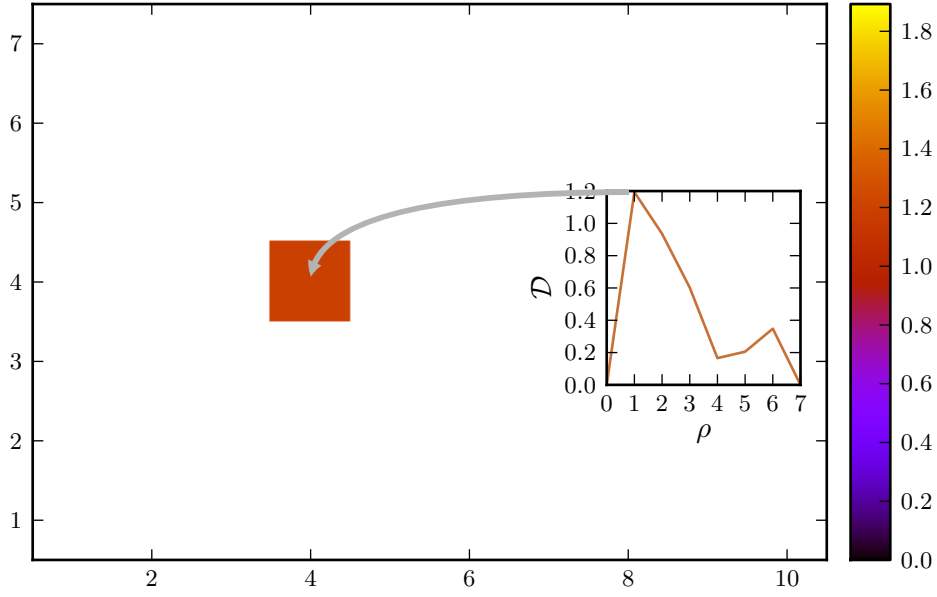


Figure 4.3: Contribution of the highlighted region of figure 4.2 to the Minkowski map. The maximum deviation strength of the local structure determines the value of the Minkowski map at the center of the highlighted region. For exact results the search for the maximum in the mapping of deviation strength over threshold has to be accounted for by trial factors (cf. section 4.1).

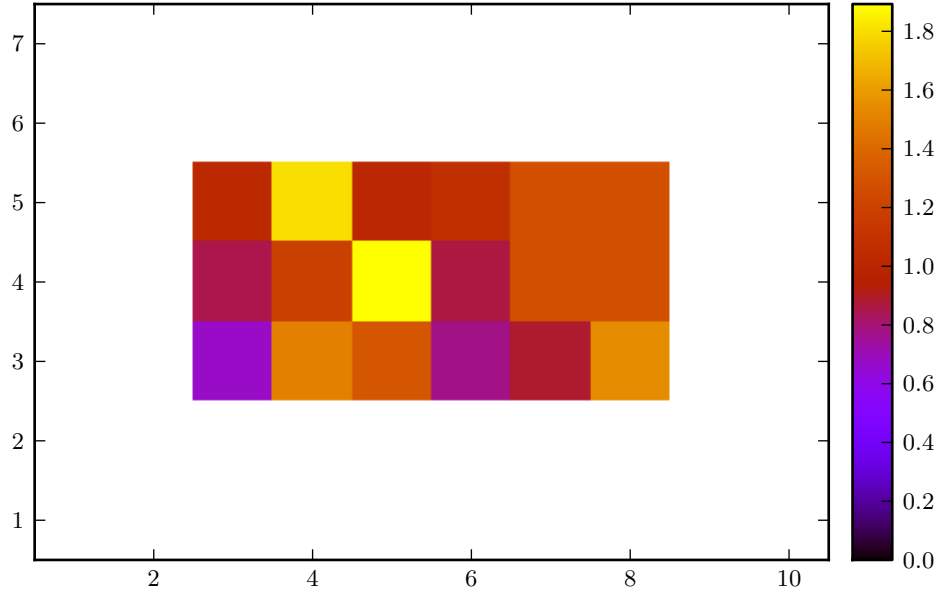


Figure 4.4: Minkowski map of the counts map shown in figure 4.1. The values of all pixels are determined as shown in figure 4.3. The resulting Minkowski map is smaller than the initial image, but characterizes the entire region, as each pixel in the Minkowski map describes the local structure of an entire  $5 \times 5$  pixels window of the initial counts map. Due to overlapping scan regions in the counts map the shown pixels of the Minkowski map are correlated. The range of values up to 1.8 is in good agreement with the hypothesis of a background measurement (cf. table 4.1).

values shown in this Minkowski map with appropriate trial factors (cf. section 4.1 for details). As it is very interesting in the analysis of gamma-ray counts maps to see if a given deviation is the result of increased flux compared to the expected background level or the result of missing events, it is possible to add a sign to the otherwise strictly non-negative deviation strength. In section 3.3.3 it is shown that the probability distribution of a set of Minkowski tensors depends on the boundary conditions, the probability  $p$  of a pixel to be black, the number  $h$  of variable black pixels and the total number  $M$  of variable pixels. Using these values it is easily possible to augment the deviation strength with a sign via

$$\mathcal{D} = \begin{cases} \mathcal{D}_{\text{unsigned}}, & \text{if } h \geq pM \\ -\mathcal{D}_{\text{unsigned}}, & \text{if } h < pM, \end{cases} \quad (4.10)$$

where  $\mathcal{D}_{\text{unsigned}}$  is the original unsigned deviation strength. This convention allows to distinguish absorbing objects from emitting objects directly in the Minkowski map and to identify wrongly estimated background levels more easily. The Minkowski map resulting from this sign augmentation is shown in figure 4.5. As sign-augmented Minkowski maps allow very intuitive interpretations of the depicted local deviations, all following Minkowski maps use this convention. Thus, negative deviation strength never indicates a compatibility larger than 1 or similar problems, but always just indicates the “direction” of the deviation.

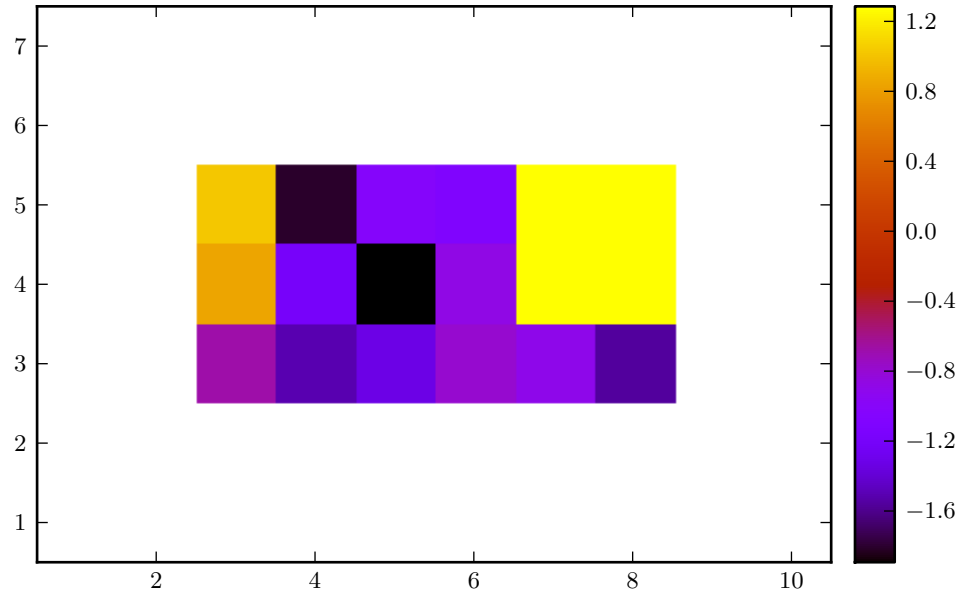


Figure 4.5: Minkowski map of figure 4.4 augmented with signs via equation (4.10). The sign of the shown deviation strength indicates the “direction” of the deviation, as explained in the text. The absolute values of the shown deviation strength correspond to the values of the unsigned Minkowski map in figure 4.4.

## 5 Background Correction

While the framework developed so far works nicely for simulated data, real data rarely fits the hypothesis of a homogeneous and isotropic background. Effects like nonuniform exposure or direction dependencies of the detector distort even the most symmetric underlying processes and lead to additional structures in the measured data. Also, there may be additional inhomogeneous background components in conflict with the most fundamental assumptions of the developed framework. It turns out that both — detector effects as well as nonuniform background components — can be corrected for in a uniform manner.

First you need to develop a model predicting the number of expected counts of every bin of the counts map. This model is the sum of all background components convolved with the exposure of the observation and all relevant detector effects. As the involved background components and detector effects vary greatly among different experiments, there is no general rule how to build such a model, but usually the needed input is required independently of the utilization of Minkowski tensors for any kind of analysis and therefore known to the corresponding experts of the experiment (examples of how to apply the structure analysis to data of the H.E.S.S. telescope array and to data of the Fermi satellite are given in chapter 6).

A straightforward way to “flatten” a given measurement is given by dividing the data by the model. The resulting scaled counts map fluctuates around 1 in every bin of a background measurement. Unfortunately, this simple approach leads to fractional counts and destroys the Poisson structure of the data. In order to quantify the fluctuations in such a scaled map you still need to know the scale factors<sup>1</sup> and thus this approach does not remove the model inhomogeneities from the data, it merely divides them among two maps, the scaled map and the model map, and you need to process both in your analysis

---

<sup>1</sup>A fluctuation of  $\pm 1$  in the original counts map may be scaled to a huge fluctuation in the scaled map if the model expectation in the given bin is small; still the fluctuation is the smallest possible deviation from the model prediction and rarely highly significant.

to be able to compute significances.

It is, however, possible to retain the Poisson structure of the data and account for the inhomogeneities by artificially adjusting the exposure of the observation. Reducing the exposure of regions with high model expectations and increasing the exposure of regions with low model expectations leads to an overall homogeneous and isotropic data set. How to achieve this without introducing artificial structures into the data and without requiring additional observations is explained in the following sections.

## 5.1 Exposure Reduction via Post-Selection

The exposure of high intensity regions can easily be reduced by randomly discarding a certain fraction of the measured events. If the model prediction of a given bin is  $\lambda_m$  and the targeted reduced level is  $\lambda_f$ , then the fraction of events to be discarded is given by  $q = 1 - \lambda_f/\lambda_m$ . Equivalently you need to retain the fraction  $p = \lambda_f/\lambda_m$  of the events in the current bin. To find the exposure-corrected count of this bin you may generate a random number sampled from the binomial distribution

$$\mathcal{P}_b(k; N, p) = \binom{N}{k} p^k (1 - p)^{N-k}, \quad (5.1)$$

where  $k$  is the corrected count value,  $N$  the original count value and  $p$  the ratio of the target level and the model expectation.

If the initial count is Poisson distributed, i.e.

$$\mathcal{P}_p(N; \lambda_m) = \frac{\lambda_m^N}{N!} e^{-\lambda_m}, \quad (5.2)$$

then the distribution of the reduced value is given by

$$\begin{aligned}
 \mathcal{P}_{\text{reduced}}(k) &= \sum_{N=0}^{\infty} \mathcal{P}_p(N; \lambda_m) \mathcal{P}_b(k; N, p) \\
 &= \sum_{N=0}^{\infty} \frac{\lambda_m^N}{N!} e^{-\lambda_m} \binom{N}{k} p^k (1-p)^{N-k} \\
 &= e^{-\lambda_m} \left( \frac{\lambda_f}{\lambda_m} \right)^k \sum_{N=k}^{\infty} \frac{\lambda_m^N}{N!} \frac{N!}{k!(N-k)!} \left( 1 - \frac{\lambda_f}{\lambda_m} \right)^{N-k} \\
 &= e^{-\lambda_m} \frac{\lambda_f^k}{\lambda_m^k} \sum_{N'=0}^{\infty} \frac{\lambda_m^{N'+k}}{k!N'!} \left( 1 - \frac{\lambda_f}{\lambda_m} \right)^{N'} \\
 &= e^{-\lambda_m} \frac{\lambda_f^k}{\lambda_m^k} \frac{\lambda_m^k}{k!} \sum_{N'=0}^{\infty} \frac{\lambda_m^{N'} \left( 1 - \frac{\lambda_f}{\lambda_m} \right)^{N'}}{N'!} \\
 &= e^{-\lambda_m} \frac{\lambda_f^k}{k!} e^{\lambda_m - \lambda_f} \\
 &= \mathcal{P}_p(k; \lambda_f),
 \end{aligned} \tag{5.3}$$

i.e. the reduced value is Poisson distributed with the targeted expectation value  $\lambda_f$ . Thus, random sampling from a binomial distribution based on the actual measurement can be used to reduce the model expectation, while preserving the Poisson structure of the data.

Detailed studies by Klatt (2010) have shown that this approach works very well. It does not introduce artificial structures and may even be used to remove known gamma-ray sources from a given data set to study weak extended sources which might otherwise be shadowed by strong point sources. As this post-selection of events reduces the effective exposure of the given bin, it also reduces the sensitivity compared to a full structure analysis of the unmodified data, but it allows to use the efficient algorithms developed in the previous chapters for the analysis of homogeneous and isotropic background models.

## 5.2 Exposure Gain via Monte Carlo Observations

While it is easily feasible to reduce the effective exposure by a certain fraction, it seems impossible to artificially increase the effective exposure without

performing additional measurements to increase the actual exposure. Nevertheless, you may perform a Monte Carlo measurement as a viable alternative to performing an additional measurement to increase the effective exposure of a given bin. If the model prediction of this bin is  $\lambda_m$  and the targeted increased level is  $\lambda_f$ , you may add a random number of counts  $\Delta k$  to the measured value  $N$ , where  $\Delta k$  is Poisson distributed with an expectation value of  $\lambda_f - \lambda_m$ . If the initial count  $N$  is Poisson distributed with an expectation value of  $\lambda_m$ , the resulting Poisson filled count  $k = N + \Delta k$  is distributed like

$$\begin{aligned}
 \mathcal{P}_{\text{filled}}(k) &= \sum_{N=0}^k \mathcal{P}_p(N; \lambda_m) \mathcal{P}_p(k - N; \lambda_f - \lambda_m) \\
 &= \sum_{N=0}^k \frac{\lambda_m^N}{N!} e^{-\lambda_m} \frac{(\lambda_f - \lambda_m)^{k-N}}{(k - N)!} e^{-\lambda_f + \lambda_m} \\
 &= e^{-\lambda_f} \sum_{N=0}^k \frac{\lambda_m^N}{N!} \frac{(\lambda_f - \lambda_m)^{k-N}}{(k - N)!} \\
 &= e^{-\lambda_f} \sum_{N=0}^k \frac{k!}{k! N! (k - N)!} \lambda_m^N (\lambda_f - \lambda_m)^{k-N} \\
 &= e^{-\lambda_f} \frac{1}{k!} \sum_{N=0}^k \binom{k}{N} \lambda_m^N (\lambda_f - \lambda_m)^{k-N} \\
 &= e^{-\lambda_f} \frac{1}{k!} (\lambda_m + \lambda_f - \lambda_m)^k \\
 &= \mathcal{P}_p(k; \lambda_f),
 \end{aligned} \tag{5.4}$$

i.e. the filled value is Poisson distributed with the targeted expectation value  $\lambda_f$ . Thus, adding random samples from a Poisson distribution can be used to increase the model expectation, while preserving the Poisson structure of the data.

Detailed studies by Göring (2008) have shown that this approach works very well. It does not introduce artificial structures and can be used to compensate for inhomogeneities of the measured data. As source signals in filled bins are covered by a layer of pure background Monte Carlo events, this method also reduces the sensitivity compared to a full analysis of the unmodified measurement, but it allows to use the much simpler background model of homogeneous and isotropic Poisson counts and the efficient algorithms introduced before.



Combining both techniques, the post-selection and the Poisson fill, it is possible to minimize their negative impact on the sensitivity of the subsequent analysis. Let  $\lambda_{m,\max}$  be the maximum of the predicted model counts and  $\lambda_{m,\min}$  the respective minimum. Instead of using post-selection to reduce all data to the value  $\lambda_f = \lambda_{m,\min}$  and thereby discarding a large fraction of events, or using Poisson filling to increase all data to the value  $\lambda_f = \lambda_{m,\max}$  and thereby covering some regions with a substantial layer of background events, you may choose an arbitrary flat value

$$\lambda_f = l\lambda_{m,\max} + (1 - l)\lambda_{m,\min} \quad (5.5)$$

with an arbitrary background level  $l \in [0; 1]$  and use Poisson filling for all bins where  $\lambda_m < \lambda_f$  and post-selection for all bins, where  $\lambda_m > \lambda_f$ . This way you can tune the desired background level  $l$  to the mean background level of your region of interest and thus run an analysis of mainly unmodified data in your region of interest with full sensitivity and you can still see all the surrounding regions, albeit with reduced sensitivity.

### 5.3 Robust Fit of Background Models

In most cases it is not feasible to predict the expected counts of each bin of a sky map analytically. Usually the model prediction depends on some parameters  $\vec{\theta}$  and these parameters have to be estimated from the data via fitting the model to the data. If there are additional signals in the data set, like unknown gamma-ray sources, this fit is distorted. Thus, you have to carefully exclude all potential source regions from the background model fit. Manually excluding source regions is error prone and in the presence of unknown sources it might even lead to a significant overestimation of the background and prevent the detection of these sources. This section introduces a robust fit algorithm, which does not rely on manually selected exclusion regions, but automatically characterizes the input data points as trustworthy or outliers during the fit and adjusts the model to the trustworthy points, ignoring potential outliers.

For a 1-dimensional data set figure 5.1 shows the influence of outliers on the maximum-likelihood fit, the least squares fit and the robust fit introduced here of a simple model. While the true parameters of the background model are  $\mu = 0$ ,  $\sigma = 10$  and  $N = 10000$ , the maximum-likelihood parameters

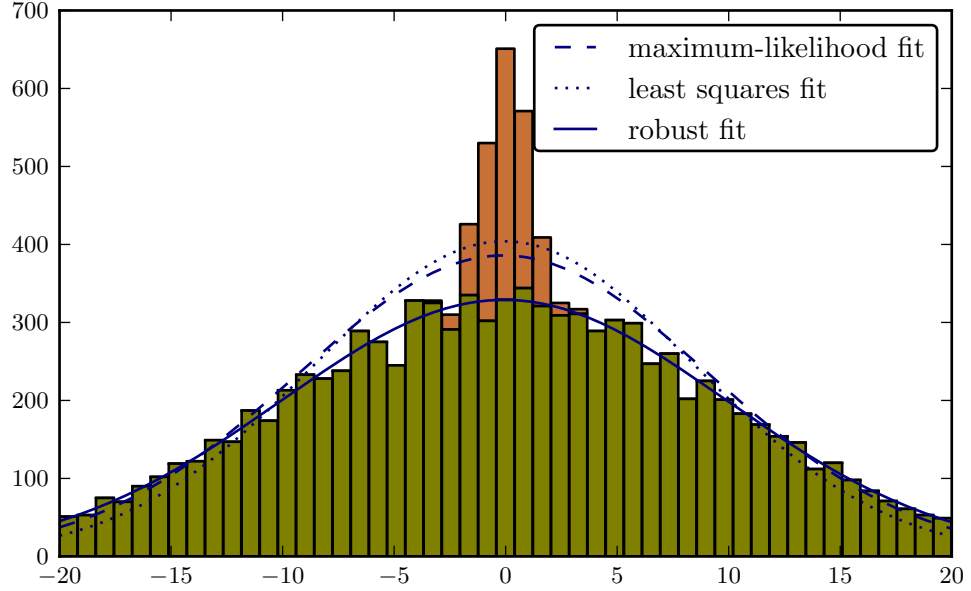


Figure 5.1: Influence of outliers on different fit algorithms. The histograms show simulated data sampled from a Gaussian distribution with  $\mu = 0$ ,  $\sigma = 10$  and  $N = 10000$  (green) and  $\mu = 0$ ,  $\sigma = 1$  and  $N = 1000$  (brown), where  $N$  is the number of sampled events. The brown histogram therefore acts as an additional source component on top of the background given by the green histogram. This additional contribution leads to outliers if only the green component is modeled. The plotted functions show best fit approximations of the combined data using a single Gaussian component. The maximum-likelihood fit and the least squares fit are clearly distorted by the outliers in the data set, while the robust fit automatically excludes them from the fit and therefore yields the best approximation to the underlying green histogram.

( $\mu = -0.10 \pm 0.1$ ,  $\sigma = 9.2 \pm 0.1$  and  $N = 10904 \pm 105$ ) and the least squares parameters ( $\mu = -0.06 \pm 0.4$ ,  $\sigma = 8.5 \pm 0.4$  and  $N = 10600 \pm 447$ ) are clearly distorted by the additional component in the data set. The robust parameter estimation ( $\mu = -0.08 \pm 0.1$ ,  $\sigma = 10.0 \pm 0.1$  and  $N = 10102 \pm 152$ ) on the other hand is in good agreement with the true background parameters.

This robust fit is essentially based on a usual maximum-likelihood fit. Using the maximum-likelihood approach you have to minimize the function

$$\mathcal{L}(\vec{\theta}) = - \sum_i \ln p_i(\vec{\theta}), \quad (5.6)$$

with respect to  $\vec{\theta}$  in order to derive an estimate  $\vec{\theta}_0$  of the parameters based on some binned data, where  $p_i(\vec{\theta})$  denotes the probability to observe the given count in the  $i^{\text{th}}$  bin, assuming the parameters  $\vec{\theta}$  are the true parameters of the model. The location of this minimum does not change when  $\mathcal{L}$  is multiplied by an arbitrary positive constant, so instead of minimizing  $\mathcal{L}$  you may minimize

$$\mathcal{L}'(\vec{\theta}) = -\frac{1}{N} \sum_i \ln p_i(\vec{\theta}), \quad (5.7)$$

where  $N$  is the number of bins. Hence, the maximum-likelihood approach minimizes the mean contribution of every bin to  $\mathcal{L}$ . Instead of estimating this mean contribution from the entire data set, which might include outliers and distort the result, more robust approaches like trimmed means or the median could be used to compute a robust estimate of this mean (cf. Huber, 1981). Minimizing this robust estimate with respect to  $\vec{\theta}$  yields a robust estimate of the parameters  $\vec{\theta}_0$  of the underlying model. While the median and trimmed means are general-purpose estimators and account for outliers in both directions,  $p_i$  provides an intrinsic measure of the deviation of the current bin from the given model. The larger  $p_i$ , the better the agreement of the current bin with the model parameters  $\vec{\theta}$ . Thus, trimming small  $p_i$  from the estimation eliminates outliers and focuses on bins that agree well with the given model. Therefore, this one-sided trimmed mean is better suited for the current purpose than general-purpose estimators like the median. So, be  $\Xi(\rho; \vec{\theta}) = \{p_i(\vec{\theta}) | p_i(\vec{\theta}) \geq \rho\}$  the set of all  $p_i(\vec{\theta})$  at least as large as  $\rho$ , then minimizing

$$\tilde{\mathcal{L}}(\vec{\theta}; \rho) = -\frac{1}{|\Xi(\rho; \vec{\theta})|} \sum_{p \in \Xi(\rho; \vec{\theta})} \ln p \quad (5.8)$$

yields a robust estimate of the parameters  $\vec{\theta}$ . For  $\rho = 0$  this reproduces the usual maximum-likelihood fit, as  $\tilde{\mathcal{L}}(\vec{\theta}; 0) = \mathcal{L}'(\vec{\theta})$ , but for larger  $\rho$  it excludes the bins which do not agree well with the current model. As the set of bins which contribute to  $\tilde{\mathcal{L}}$  depends on  $\vec{\theta}$ , the mean bin contribution to the overall likelihood is always estimated from the set of bins fitting the current set of parameters  $\vec{\theta}$  best. Thus, globally optimizing  $\tilde{\mathcal{L}}(\vec{\theta}; \rho)$  with respect to  $\vec{\theta}$  results in a fit of the model parameters to a subset of all the bins which agrees well with the given model.

Unfortunately the optimum  $\rho$  to achieve robustness with respect to outliers and still use the maximum information available from the reliable subset of the data depends on the model and the input data. A more universal quantity to tune the robustness of the fit is given by the maximum fraction of outliers expected to be in the data. This fraction is denoted by  $\epsilon$  in the following and the fraction of reliable data is respectively given by  $\alpha = 1 - \epsilon$ . It is usually straightforward to estimate  $\epsilon$  for a given observation. Knowing  $\epsilon$ ,  $\Xi_\epsilon(\vec{\theta})$  should be chosen such that

$$|\Xi_\epsilon(\vec{\theta})| \approx \alpha N, \quad (5.9)$$

to ensure that a maximum amount of reliable data is used to estimate the parameters, while potential outliers are ignored. Let  $p_{\{k\}}(\vec{\theta})$  be the  $k^{\text{th}}$  entry of a tuple of all  $p_i(\vec{\theta})$ , sorted in decreasing order, then choosing  $\rho = p_{\{\lfloor \alpha N \rfloor\}}(\vec{\theta})$ , i.e.

$$\Xi_\epsilon(\vec{\theta}) = \Xi(p_{\{\lfloor \alpha N \rfloor\}}(\vec{\theta}); \vec{\theta}), \quad (5.10)$$

fulfills these conditions. It chooses the  $\lfloor \alpha N \rfloor$  largest  $p_i(\vec{\theta})$  as the set of trustworthy bins for the current model parameters  $\vec{\theta}$ . In addition to these bins it also includes all bins in the fit which are just as good as the worst bin of the trustworthy set. Everything else is considered a potential outlier and ignored in the estimation of the mean likelihood. So, just like for  $\Xi(\rho; \vec{\theta})$ , the size of  $\Xi_\epsilon(\vec{\theta})$  and the bins contributing to the set depend on  $\vec{\theta}$ , but in contrast to  $\Xi(\rho; \vec{\theta})$  for a fixed  $\rho$  the size of  $\Xi_\epsilon(\vec{\theta})$  is always at least  $\lfloor \alpha N \rfloor$ . When minimizing

$$\tilde{\mathcal{L}}(\vec{\theta}; \epsilon) = -\frac{1}{|\Xi_\epsilon(\vec{\theta})|} \sum_{p \in \Xi_\epsilon(\vec{\theta})} \ln p \quad (5.11)$$

with respect to  $\vec{\theta}$  the bins contributing to  $\Xi_\epsilon(\vec{\theta}_0)$  at the location of the minimum  $\vec{\theta}_0$  represent a subset of the data which agrees well with the given model, and  $\vec{\theta}_0$  is the maximum-likelihood fit of the model to this subset of the data.

This simultaneous optimization of the model parameters and the subset of bins to use for the fit can not only be used to achieve robustness with respect to outliers, but also to prevent numerical issues. By automatically classifying bins with problematic floating point contributions like *NaN* (not a number) or *Inf* (infinity) as outliers, the robust fit can handle start parameters which lead to numerical problems with a normal maximum-likelihood fit of the entire data set.

It might seem that choosing the bins which fit best to the model for each set of parameters  $\vec{\theta}$  biases the optimization towards the start values used for the fit. The mean likelihood of the starting step is estimated from bins fitting the start values and subsequent likelihoods are estimated from bins fitting the subsequent parameters chosen by the minimization algorithm. As long as  $\epsilon < 0.5$  this bias is more of a secondary concern, as the fit has to find parameter values which fit the majority of the data and can not isolate an arbitrary subset of bins fitting the starting values. Figure 5.2 shows some intermediate steps of a robust fit to some data containing outliers. While its

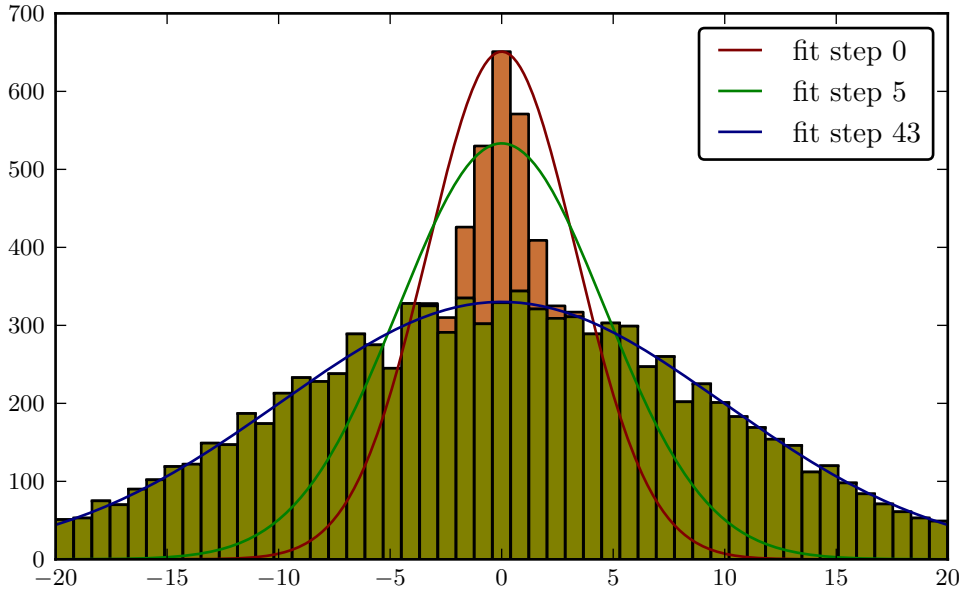


Figure 5.2: Some intermediate steps of the robust fit of figure 5.1

starting values favor the inclusion of many outliers in the set of bins to fit, the majority of the data does not fit this estimate and pulls the parameters towards the real model parameters.

All the advantages of the robust fit, like robustness with respect to outliers and with respect to numerical issues, come with a price. Using only a subset of the available data to determine the model parameters increases the statistical uncertainties of the estimate. Assuming a  $1/\sqrt{N}$ -scaling of the statistical fluctuations, the fluctuations of the robust fit are a factor of  $1/\sqrt{\alpha}$  larger than the fluctuations of a fit to the entire data set. For a data set contaminated with about 10% of outliers this corresponds to  $1/\sqrt{0.9} \approx 1.054$ , i.e. a 5% increase of the statistical fluctuations of the estimated value and for the extreme case of  $\epsilon = 50\%$  you find an increase of about 41% of the statistical fluctuations. Thus, the increase of statistical fluctuations due to the removal of potential outliers is smaller than the fraction of contaminated data. This is a small price to pay for the removal of all potential outliers compared to the advantages of removing the bias introduced by these outliers on the estimated parameters.

A more severe limitation of the robust fit comes from the shape of the function which needs to be minimized. While the negative log likelihood or the sum of squared residuals are smooth functions if the underlying model is sufficiently smooth, the robust quantifier  $\tilde{\mathcal{L}}(\vec{\theta}; \epsilon)$  is only piecewise smooth, as shown in figure 5.3. This corresponds to the fact that  $\tilde{\mathcal{L}}(\vec{\theta}; \epsilon)$  is a piecewise combination of all possible negative log likelihood functions of all possible subsets  $\Xi_\epsilon(\vec{\theta})$  of the data. Therefore, you should not rely on gradient algorithms to minimize  $\tilde{\mathcal{L}}(\vec{\theta}; \epsilon)$  but rather resort to more general minimization algorithms like the downhill simplex method or simulated annealing (Press, 2007), which are usually significantly slower than gradient algorithms. In addition to destroying global smoothness, the piecewise structure may easily introduce additional local minima if several of the contributing pieces contain the minimum of the corresponding negative log likelihood function of the current data subset. While local minima are a general problem of nonlinear parameter estimation and should always be considered when minimizing the corresponding quantifier, the location of the additional local minima of  $\tilde{\mathcal{L}}(\vec{\theta}; \epsilon)$  is very unfortunate. With non-robust fits it is often possible to resolve the issue of local minima by a suitable choice of starting values of the fit, motivated from the physics background of the model. The local minima of  $\tilde{\mathcal{L}}(\vec{\theta}; \epsilon)$ , on the other hand, tend to cluster at the real parameter values and

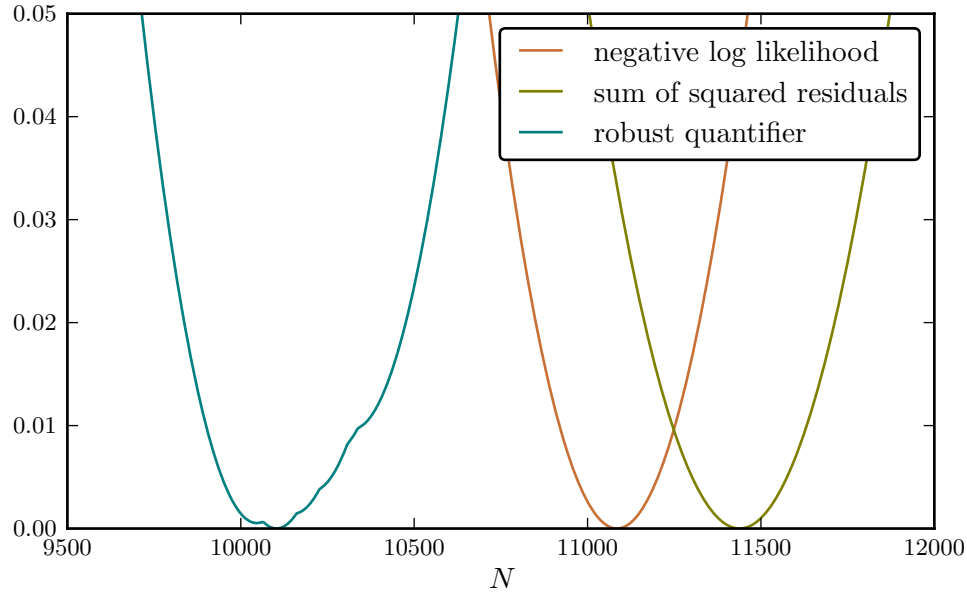


Figure 5.3: The three functions show a slice through the parameter space of the three multivariate functions which need to be minimized to find the best fit values of the fits shown in figure 5.1. The negative log likelihood corresponds to the maximum-likelihood fit, the squared residuals to the least squares fit and the robust quantifier to the robust fit respectively.  $\mu$  and  $\sigma$  are set to the true values of the underlying model and the dependency on  $N$  is shown. The function values are given in arbitrary units, as each function is scaled and shifted to the same range of values for comparison.

thus can not be avoided by starting the fit close to the expected parameters. This clustering is due to the location of the global minima of the negative log likelihood functions of the different subsets. If a subset contains only a few mild outliers its global minimum is close to the real parameters and therefore close to the global minimum of  $\tilde{\mathcal{L}}(\vec{\theta}; \epsilon)$ . To alleviate this issue you should restart the simplex algorithms several times with a large simplex centered at the last found minimum or resort to simulated annealing.

The robust fit provides a convenient method to determine parameters of background models in gamma-ray astronomy without having to care for potential additional sources in the measured data. Using the background model and the parameters from the robust fit, it is easily possible to compute Minkowski maps for a wide range of different experiments. The next chapter gives examples of applying the developed analysis framework consisting of a robust background model fit and a detailed structure analysis of the data using Minkowski tensors.



## 6 Application to Gamma-Ray Astronomy Data

After a detailed discussion of the theoretical foundations in the previous chapters this chapter shows how to apply the morphometric analysis to real data of gamma-ray experiments. The robust fit (section 5.3) is used to fit detector-specific background models (including exposure and point spread function effects) to the data and to estimate the parameters of this model in a robust way with respect to the influence of potential sources in the given data set. The background model is used to transform the measured data into a data set with homogeneous and isotropic Poisson background (sections 5.1 and 5.2). Minkowski tensors are used to quantify the structure of this transformed data set (chapter 2) and the full probability distribution of these Minkowski tensors for homogeneous and isotropic Poisson background (chapter 3) is used to quantify the compatibility of the measured structure tensors with the typical structure of the background and to compute Minkowski maps visualizing this compatibility locally (chapter 4).

In the following section the structure analysis is applied to data taken with the H.E.S.S. experiment. After examining the general capabilities and characteristics of the analysis in this context, a short discussion of the analysis of Fermi data demonstrates the flexibility and limits of the proposed technique.

### 6.1 The H.E.S.S. Experiment

The High Energy Stereoscopic System (H.E.S.S.) is an array of four imaging air Cherenkov telescopes for the detection of cosmic gamma rays of energies above 100 GeV. Its field of view (FoV) is  $5^\circ$  in diameter and the 68% containment radius of the point spread function (PSF) is about  $0.1^\circ$  (F. Aharonian et al., 2006). The telescope array records the images of electromagnetic air showers induced by the high-energy gamma rays in the atmosphere by collecting the Cherenkov light emitted by relativistic particles in these showers. The

stereoscopic observation of air showers, i.e. their simultaneous observation from different locations respectively telescopes, facilitates the reconstruction of the shower direction and origin. The reconstructed shower direction gives direct feedback on the initial direction of the incident gamma ray. As gamma rays are electrically neutral and therefore not affected by interstellar magnetic fields, their direction information can be used to track the gamma ray back to its cosmic origin.

Unfortunately, gamma rays are not the only cosmic particles causing atmospheric air showers. Actually, the largest fraction is caused by charged particles, mainly by protons. Sophisticated trigger conditions and selection cuts on the recorded camera images reduce this background for H.E.S.S. significantly. Still, even after these background rejection methods a considerable amount of recorded events is due to charged cosmic rays and not due to gamma rays. As these charged cosmic rays lose their correlation with the direction of their origin in interstellar magnetic fields on their way from the source of the cosmic rays to Earth, this remaining background can be assumed to be homogeneous and isotropic. Thus, gamma-ray sources can be detected as localized excess of gamma-like events on top of a homogeneous and isotropic flux of background events.

A more detailed discussion of the basic detection and analysis principles used by the H.E.S.S. collaboration is given in F. Aharonian et al. (2006). An overview of some more recent developments and advances in event reconstruction techniques and analysis methods, like the use of the high performance likelihood reconstruction of de Naurois and Rolland (2009) or advanced models of the telescope acceptance, is e.g. given in de Naurois (2012).

### 6.1.1 Background Model

In the H.E.S.S. jargon an acceptance map is a sky map reflecting the spatial distribution of the detection efficiency of the telescope array. The peak value of an acceptance map is usually set to 1 by dividing all bins of the unscaled acceptance sky map by the peak value of this unscaled map. The resulting acceptance map shows the fraction of detected gamma-like events in each bin of the sky map compared to the peak performance of the telescope array. So, if a bin of the acceptance map shows a value of 0.3, only 30% of the events which could have been detected in the peak performance region are detected in this bin of the sky map.

Such acceptance maps are usually the result of combining many individual observations with different observation conditions and pointing positions, which leads to effects like an increased acceptance in regions where the different observations overlap and reduced acceptance in observations with bad atmospheric conditions compared to observations where the air showers can be observed on a perfectly clear night sky. In addition to these exposure effects, the acceptance of a single observation is dominated by the spatial sensitivity of the telescopes to air showers. As it gets increasingly difficult to reconstruct the properties of the initial gamma ray if the recorded showers are close to the edges of the FoV, a larger fraction of these events is discarded by data quality selection cuts. This leads to an approximately radially symmetric acceptance profile of the telescopes where the peak of this profile is at the pointing position in the center of the FoV and the acceptance decreases towards the edges of the FoV. A more detailed discussion of acceptance maps and their determination is given in de Naurois (2012). This work also discusses the so called 2D acceptance model, an advanced technique which does not rely on the radial symmetry of the telescope acceptance and thus accounts for higher-order effects like broken pixels in the cameras. Since the technical details are irrelevant to the basic concepts of structure analysis discussed here, the following paragraphs just use the acceptance maps created from the 2D acceptance model and you are referred to de Naurois (2012) for details on the underlying algorithm.

As the acceptance map incorporates all detector and observation effects like exposure, spatial detection efficiency and observation conditions, the acceptance map and a counts map of the reconstructed gamma-like events are all you need to start the structure analysis. The corresponding background model describing the expected background of the counts map is simply given by

$$\lambda_{ij} = \lambda \alpha_{ij}, \quad (6.1)$$

where  $\lambda_{ij}$  is the expected number of counts in bin  $(i, j)$ ,  $\alpha_{ij}$  the acceptance level of this bin and  $\lambda$  the global background level of the underlying homogeneous and isotropic Poisson background caused by air showers induced by charged cosmic rays. Thus, the probability  $p_{ij}$  to find  $k_{ij}$  background events in bin  $(i, j)$  is given by

$$p_{ij} = \frac{\lambda_{ij}^{k_{ij}}}{k_{ij}!} e^{-\lambda_{ij}}. \quad (6.2)$$

Using these  $p_{ij}$  as input to equation (5.11), the robust fit of section 5.3 can be used to adjust the background model to the counts map in order to determine the global background level  $\lambda$  of the underlying homogeneous and isotropic background, the only free parameter of the model.

After the robust fit,  $\lambda_{ij}$  is known for all bins of the counts map. Using the post-selection method and the Poisson filling discussed in sections 5.1 and 5.2, the counts map can be transformed into a counts map with homogeneous and isotropic acceptance, i.e.  $\lambda'_{ij} = \lambda_f$ , where  $\lambda_f$  is the chosen background level and should correspond to the mean background level of the main region of interest (cf. section 5.2).

A robust fit of a flat model to this acceptance-corrected counts map can be used as self consistency check to detect systematic problems and to account for statistical fluctuations in the Monte Carlo methods of the acceptance correction. Fitting

$$p_{ij} = \frac{\lambda'^{k_{ij}}}{k_{ij}!} e^{-\lambda'} \quad (6.3)$$

to the acceptance-corrected counts map allows to check if  $\lambda' \approx \lambda_f$ . If  $|\lambda' - \lambda_f| > \sqrt{\lambda_f}$ , i.e. if the estimated background level deviates from the desired background level by more than one standard deviation of the corresponding Poisson distribution, the background model — in this case the acceptance map — and the current data set should be checked carefully for systematic errors. If both values agree well,  $\lambda'$  should be chosen to represent the background level of the acceptance-corrected counts map, because it accounts for statistical fluctuations of the Monte Carlo methods of sections 5.1 and 5.2 and thus reflects the actual mean background level of the acceptance-corrected counts map.

The acceptance-corrected counts map is turned into a sequence of b/w pixel images via thresholding (cf. section 2.1.4). The probability of a pixel to be black in such an image is given by

$$p_\rho = \sum_{k=\lceil \rho \rceil}^{\infty} \frac{\lambda'^k}{k!} e^{-\lambda'}, \quad (6.4)$$

assuming the count  $k$  of the corresponding bin of the sky map is entirely caused by background events. So, the probability of a background pixel to be black just corresponds to the probability to find a Poisson count  $k$  above the threshold  $\rho$ .

After selecting a set  $\mathcal{T}$  of Minkowski tensors to be used to quantify the structure of the acceptance-corrected counts map and some boundary conditions to evaluate these tensors at the edge of a b/w image, plugging  $p_\rho$  into equation (3.34) yields the common probability distribution of the structure functionals for the background model. After this step the methods described in chapter 4 can be used to quantify deviations from the typical background structure and to compute Minkowski maps visualizing the local structure deviations in the acceptance-corrected sky map.

### 6.1.2 The Crab Nebula, a View of the Standard Candle

The Crab Nebula is one of the strongest steady VHE gamma-ray sources in the sky. These properties made the Crab Nebula not only the first ever by a Cherenkov telescope detected VHE gamma-ray source (Weekes et al., 1989), they also allow to use it as a source to cross calibrate different experiments. Therefore, the Crab Nebula is often called the *standard candle* of VHE gamma-ray astronomy. With respect to the angular resolution of H.E.S.S. the extension of this standard candle is compatible with the PSF of the telescopes and thus can be regarded as point source.

In the following this point-like standard candle is used to show the basic properties of the structure analysis. For this, a current data set of the H.E.S.S. experiment is used, comprising a total live time of about 49 h after standard data quality selection cuts. The corresponding counts map of gamma-like events reconstructed using the likelihood method discussed in de Naurois and Rolland (2009) is shown in figure 6.1. The Crab Nebula is strong enough to be clearly visible in this sky map without need for background subtraction or any particular structure analysis. If the color scale is adjusted to show more details of the surrounding sky, the inhomogeneities of the used observations are clearly visible (cf. figure 6.2). This structure is caused by an inhomogeneous exposure of the sky and by detector effects like trigger conditions and the spatial sensitivity of the used Cherenkov cameras. Using the 2D acceptance model of de Naurois (2012), these effects are all combined into the acceptance map shown in figure 6.3.

As discussed before, this map essentially determines the background model of the given observations and  $\lambda$ , the only free parameter of the model, can be fixed using the robust fit to adjust the background model to the data shown in figure 6.1 (cf. equations (6.1) and (6.2)). As the Crab Nebula is

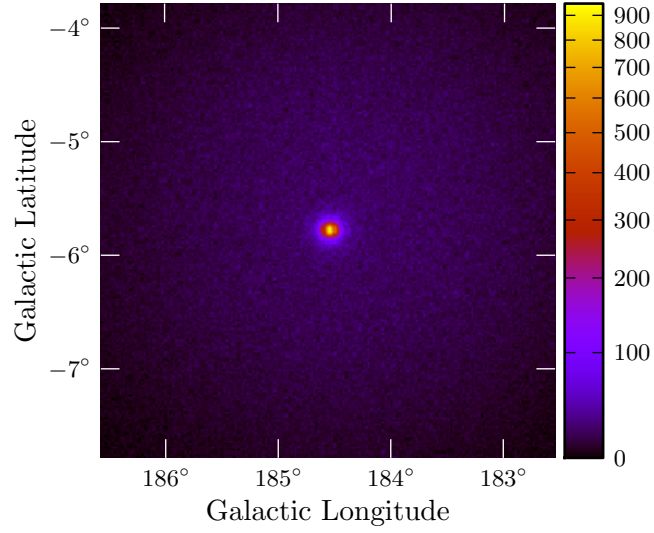


Figure 6.1: Sky map showing the counts of gamma-like events from the region around the Crab Nebula as observed with H.E.S.S. in about 49 h.

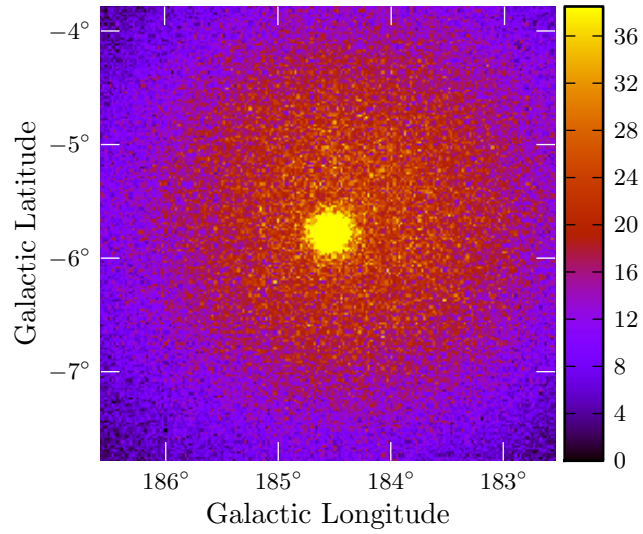


Figure 6.2: Sky map of figure 6.1 with color scale adjusted to show more details of the surroundings of the strong point source.

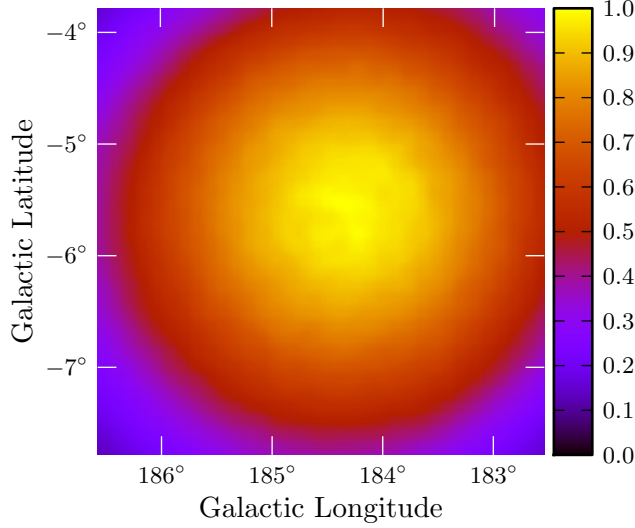


Figure 6.3: 2D acceptance model of the underlying observations of the reconstructed gamma-like events shown in figure 6.1.

significantly smaller than the observed region, it contaminates less than 10% of the bins in figure 6.1 with additional source counts. So, a robust fit with  $\epsilon = 0.1$  easily copes with the present sources when fitting the background model to the data and yields a reliable estimate of  $\lambda$ , as it uses at least 90% of the data to determine the model parameter. As the source lies within the peak region of the acceptance map, it is reasonable to choose the background level  $l$  of equation (5.5) close to 1. The resulting acceptance-corrected counts map is shown in figure 6.4. Except for the region at the Crab Nebula this acceptance-corrected counts map shows a nice homogeneous and isotropic Poisson structure. The Monte Carlo events which are used to achieve the acceptance correction are shown in figure 6.5. As intended, setting  $l = 1$  leads to negligible changes of the data in the regions of high acceptance. Thus, the data is essentially unmodified close to the Crab Nebula and acceptance correction effects are only relevant towards the edges of the observed sky.

Using  $\mathcal{T} = \{A, P, \chi\}$ , i.e. area, perimeter and Euler characteristic as structure functionals to examine the acceptance-corrected data, the compatibility of the local structure around each bin of the counts map with the background

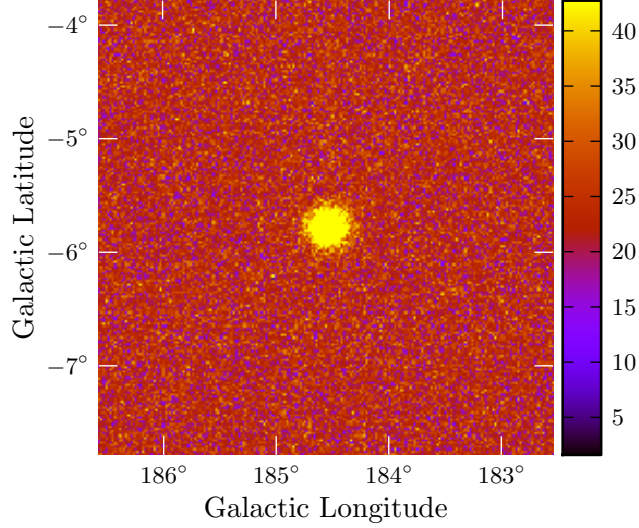


Figure 6.4: Acceptance-corrected version of figure 6.2. The methods described in chapter 5 are used to determine  $\lambda$ , the parameter of the background model, and to flatten the data.

model can be quantified. The resulting Minkowski map is shown in figure 6.6. While it is more natural to compute the deviation strength of Minkowski tensors, the unit of the significance chosen in this and all the following maps is  $\sigma$ , as this is the standard unit in VHE gamma-ray astronomy and a conversion is easily possible (cf. table 4.1). The Minkowski map shows a distinct peak at the position of the Crab Nebula and structure fluctuations compatible with background measurements everywhere else. Thus, the applied acceptance correction works and does not introduce artificial structures.

Comparing the Minkowski map with the significance map in figure 6.7, computed using standard H.E.S.S. analysis tools based on Li and Ma (1983), shows excellent agreement of both approaches. The shape and size of the source region match perfectly and the range of fluctuations in the surrounding areas are similar.

It is not surprising that both analysis approaches, the standard H.E.S.S. analysis and the morphometric analysis, are able to detect a very strong point source like the Crab Nebula. As it is not relevant whether a source



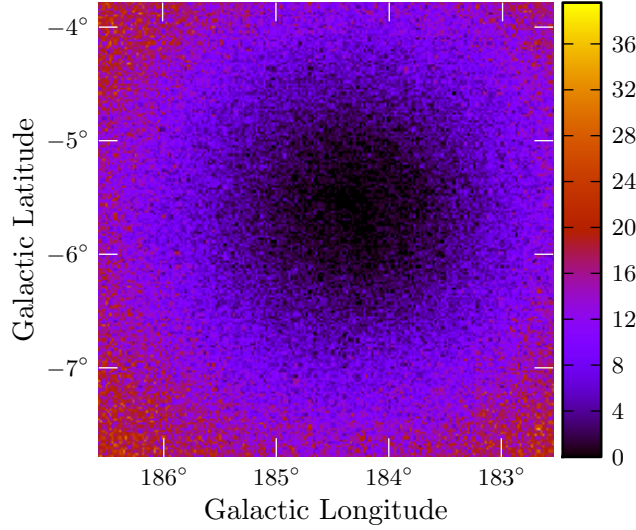


Figure 6.5: Difference of the acceptance-corrected counts map shown in figure 6.4 and the unprocessed counts map of figure 6.1. Thus, the shown counts correspond to the Monte Carlo events used to correct for acceptance effects. As these Monte Carlo events are supposed to compensate the acceptance dependencies of figure 6.1, their spatial structure resembles an inverted version of the acceptance map shown in figure 6.3.

is detected at a  $100\sigma$  or at a  $120\sigma$  significance level<sup>1</sup> the given example is also poorly suited to compare the sensitivity of both approaches. Actually, such a comparison never was the intention of this section, instead a simple first example was chosen to recapitulate the required analysis steps mixed with some hands-on examples. Still, the discussion demonstrates that the acceptance correction works and that the morphometric analysis can be used to quantify local structure deviations in order to detect gamma-ray sources via peaks in Minkowski sky maps.

Because of the small scan windows used by the structure analysis, it is questionable if larger sources than the Crab Nebula can be processed properly.

<sup>1</sup>For both analyses the actual peak significance is even higher, but at these scales it is probably sufficient to note a highly significant detection.

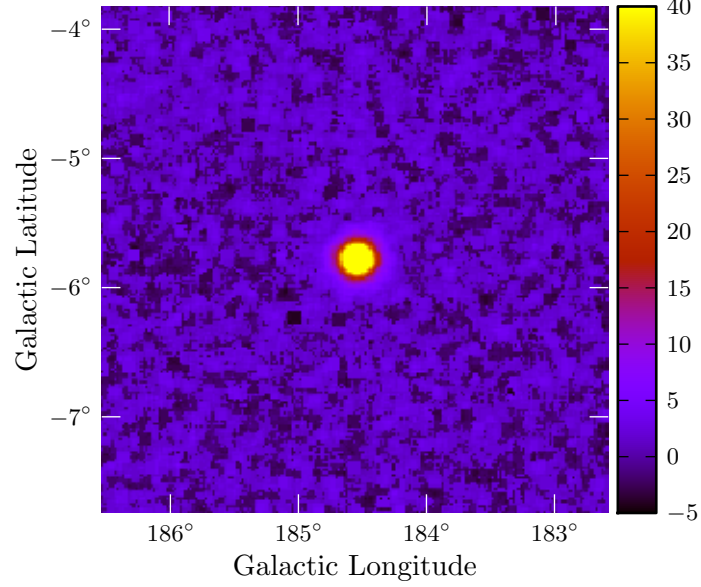


Figure 6.6: Minkowski sky map of the acceptance-corrected counts map of figure 6.4. The Minkowski tensors used to quantify the local structures of the counts map are  $\mathcal{T} = \{A, P, \chi\}$  and the structure of a  $5 \times 5$  bins scan window was used. With a bin size of  $0.02^\circ$ , this corresponds to the block-shaped artifacts found in the map. The boundary conditions chosen for the structure quantification set all pixels beyond the scan window to white. Instead of the sign-augmented deviation strength  $\mathcal{D}$  discussed in chapter 4, a sign-augmented significance in units of  $\sigma$  is shown, where the conversion from  $\mathcal{D}$  to  $\sigma$  corresponds to the mapping given in table 4.1. The significance scale is truncated at  $40\sigma$  to show more of the general structure of the map instead of focusing on the Crab Nebula peak.

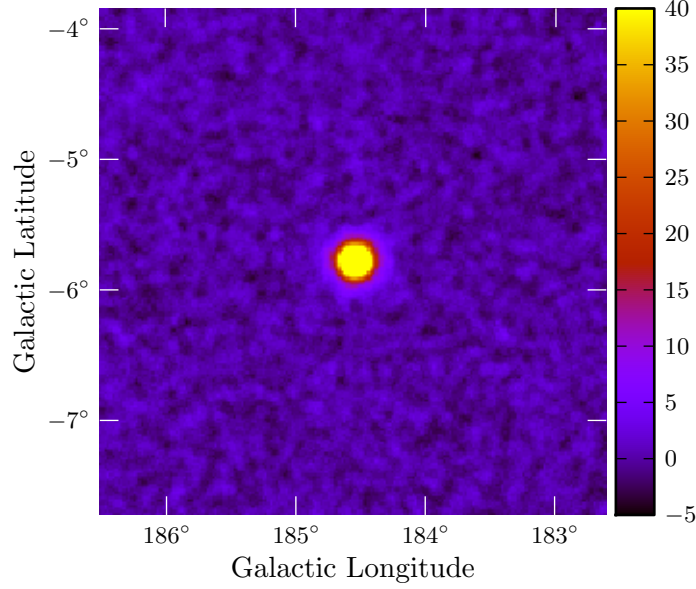


Figure 6.7: Significance map of the counts map of figure 6.1. The color code shows the significance of the events in an oversampling radius of  $0.056^\circ$  around each bin as estimated via the likelihood ratio test of Li and Ma (1983). This radius was chosen to correspond to an integration region equal in area to the area of the scan window of the Minkowski map in figure 6.6. The number of background counts needed to evaluate the significance of the counts in figure 6.1 via Li and Ma (1983) is estimated via the Template Background method (Rowell, 2003). Further details of the standard H.E.S.S. analysis used to compute this significance map are reviewed in de Naurois (2012). The color scale was truncated to correspond to the color scale of figure 6.6.

To clarify this, the next section shows that it is indeed possible to study large extended sources using Minkowski maps.

### 6.1.3 RX J1713.7–3946, a View of an Extended Source

While the last section shows that the structure analysis works for point sources, this section examines the extended source RX J1713.7–3946 to study further features of the new approach in the presence of pronounced structures.

RX J1713.7–3946 was the first extended VHE gamma-ray source whose morphology could ever be resolved by an imaging air Cherenkov telescope (Aharonian et al., 2004), and it is still one of the largest<sup>2</sup> sources of VHE gamma rays known today. Due to its size and its position in the galactic plane, close to several other VHE gamma-ray sources, the fraction of bins which might contain source events is larger than for the Crab Nebula. Otherwise, the procedure is essentially identical to the steps presented in the previous section. Therefore, only the resulting Minkowski map is shown in figure 6.8 and images of counts and acceptance maps are omitted. RX J1713.7–3946 is clearly visible in the Minkowski map along with three nearby smaller sources. This perfectly agrees with the results of the Template Background method of the standard H.E.S.S. analysis (cf. figure 6.9). Hence, it is indeed possible to process extended sources as well as smaller sources, even in the same FoV and using identical analysis parameters. While RX J1713.7–3946 is obviously a lot larger than the used scan window of  $5 \times 5$  bins, every piece of the source is locally different from the expected background structure and thus the entire source is visible in the Minkowski map in figure 6.8, where these local structure deviations are plotted in a sky map. Nevertheless, this local analysis does not provide the total significance of the entire source region, as the deviation strength of different local scan windows can not be combined to derive the global value (while Minkowski tensors are additive, the deviation strength is derived from their probability distribution and does not inherit the nice transformation properties of the underlying tensors). To derive the total significance, one would have to process the counts map using a scan window large enough to contain the entire source region. While it is computationally expensive to increase the size of the scan window, you may also adjust the

---

<sup>2</sup>RX J1713.7–3946 is one of the largest sources of VHE gamma rays with respect to its angular diameter as seen from Earth, not with respect to its spatial extension in the Universe.

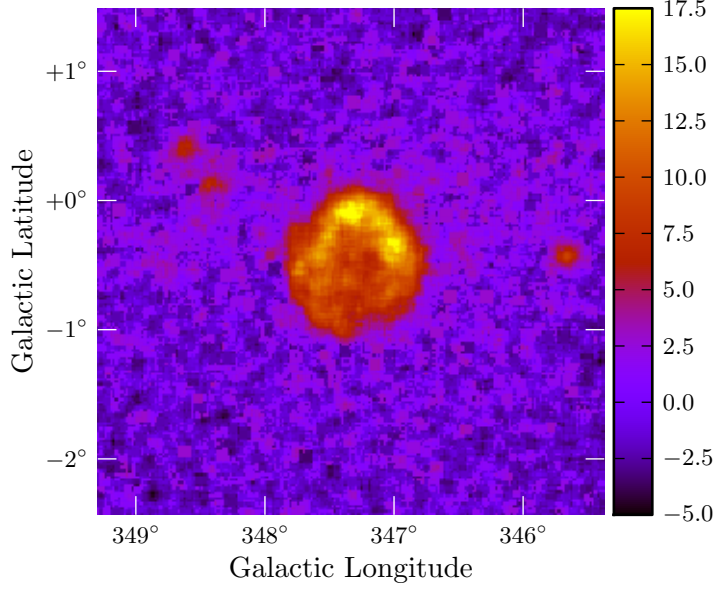


Figure 6.8: The Minkowski sky map shows the significance of local structure deviations from the background model in the region around RX J1713.7–3946. Color code and interpretation are analogous to the Minkowski map shown in figure 6.6. All analysis parameters are identical, except for the fraction of potential source bins  $\epsilon$ , which is set to 20% for RX J1713.7–3946 to account for its size and additional sources in the examined region. The underlying data set corresponds to a live time of about 107 h.

binning to make the region of interest fit into the chosen  $N \times N$  bins scan window.<sup>3</sup>

Just like in figure 6.6 for the Crab Nebula, there are some block-shaped artifacts visible in figure 6.8. These are due to single pixels with very high or very low counts compared to the expected average background level. A single extreme pixels can dominate all scan windows containing this pixel and may

<sup>3</sup>While computationally cheap, a coarser binning reduces the quantifiable structure of the counts map and therefore diminishes the advantage of Minkowski tensors compared to conventional methods. See chapter 7 for details.

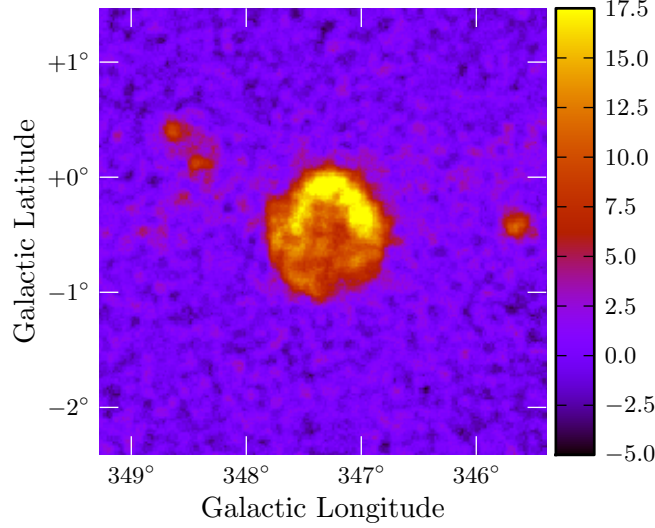


Figure 6.9: Sky map of the region around RX J1713.7–3946 showing the same observations as in figure 6.8 but analyzed using standard H.E.S.S. methods instead of the new structure analysis. The used analysis and image parameters are analogous to figure 6.7.

thus lead to an entire neighborhood of  $N \times N$  pixels in the Minkowski map reflecting the deviation strength of this single extreme pixel. Nevertheless, usually the structure of a counts map varies more smoothly and does not change from pixel to pixel. Thus, the general structure of Minkowski maps varies more smoothly and the number of block-shaped artifacts is limited. If the color scale of figures 6.8 and 6.9 is adjusted to highlight the highly significant regions of RX J1713.7–3946, both show the same fine structure of the source region, without block-shaped artifacts and on a finer scale than the  $5 \times 5$  pixel radius (cf. figure 6.10).

So far, point sources as well as extended sources have been examined. The Minkowski analysis works well in both cases and yields results compatible with standard methods. However, up to now all presented data essentially consisted of pointed observations of the discussed sources. Therefore, the sources were located close to the peak acceptance region of the map such that the required acceptance correction could be handled by filling exterior regions

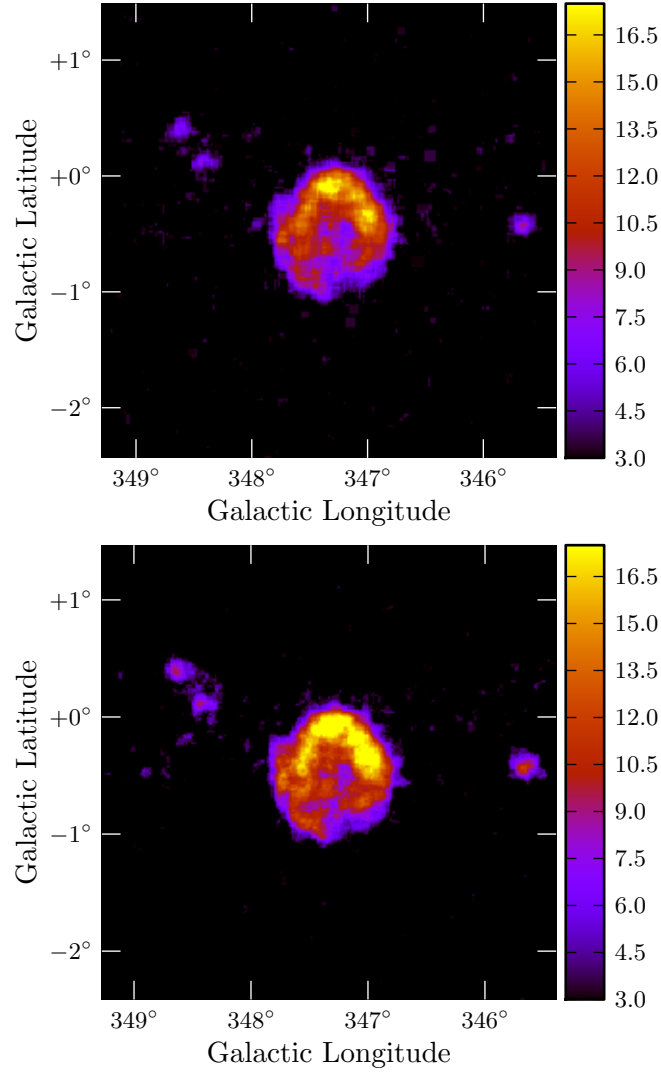


Figure 6.10: *Top*: Minkowski sky map of figure 6.8 with color scale adjusted to highlight details of the structure of the high significance regions. *Bottom*: Significance sky map of figure 6.9 with color scale adjusted to match top Minkowski sky map.

with Monte Carlo events. The next section shows an excerpt of the H.E.S.S. Galactic Scan, which involves a more complex acceptance profile and shows the full power and flexibility of the introduced methods.

#### 6.1.4 The Galactic Scan, a View of a Large Scale Analysis

Apart from pointed observations, the H.E.S.S. experiment has performed a continuous survey of the galactic plane since 2004.<sup>4</sup> This huge data set comprises dedicated survey observations at varying observation conditions as well as pointed observations of sources or source candidates in the galactic plane and adds up to a total live time of about 2139 h. The combination of this plethora of different observations leads to a quite non-uniform exposure along the plane.

As the regions of deepest exposure are around sources with lots of dedicated pointed observations, it is not representative to choose a background level of  $l = 1$  to analyze the entire plane. By reducing the exposure of these pointed observations via post-selection, as discussed in section 5.1, the required acceptance correction for all other regions becomes considerably less intrusive and the overall change to the data can be minimized. Figure 6.11 shows a Minkowski sky map of the inner region of the galactic plane for  $l = 0.5$ . The general structure of this Minkowski sky map is in very good agreement with the corresponding significance map computed using the template background method of the standard H.E.S.S. software. It shows many different sources along the galactic plane in a variety of sizes, shapes and intensities. This demonstrates that the developed framework is capable of handling huge data sets with very inhomogeneous background models and can still characterize the local structure of the entire data set, in order to find gamma-ray sources via significant deviations from this background.

Figure 6.12 shows the same data processed with a background level of  $l = 1$  for comparison. It is evident that the deviation strength of most structures is reduced compared to the analysis shown in figure 6.12. Only the significance of the galactic center region increases.<sup>5</sup> This is in perfect agreement with the expected effects of the acceptance correction. Regions of low acceptance are covered with an artificial level of background; regions of high acceptance lose

---

<sup>4</sup>See Chaves (2011) for a comprehensive review of the H.E.S.S. galactic plane survey.

<sup>5</sup>This is not visible in the given sky map, as the color scale is truncated to early to see differences in the peak values.



exposure via post-selection. Both suppresses potential source signals relative to the present background and leads to a conservative estimation of structure deviations.

Thus, the developed framework of structure analysis based on Minkowski tensors works very well for data taken with the H.E.S.S. telescopes. It can handle point sources like the Crab Nebula, large extended sources like RX J1713.7–3946 and even survey data containing many different source classes and observation conditions. The Minkowski sky maps presented thus far do not show significant improvements compared to the results of well established standard tools, but they demonstrate that the basic ideas work and that the introduced framework is applicable to data of the H.E.S.S. experiment. For details on sensitivity comparisons and further systematic studies of the morphometric analysis see chapter 7.

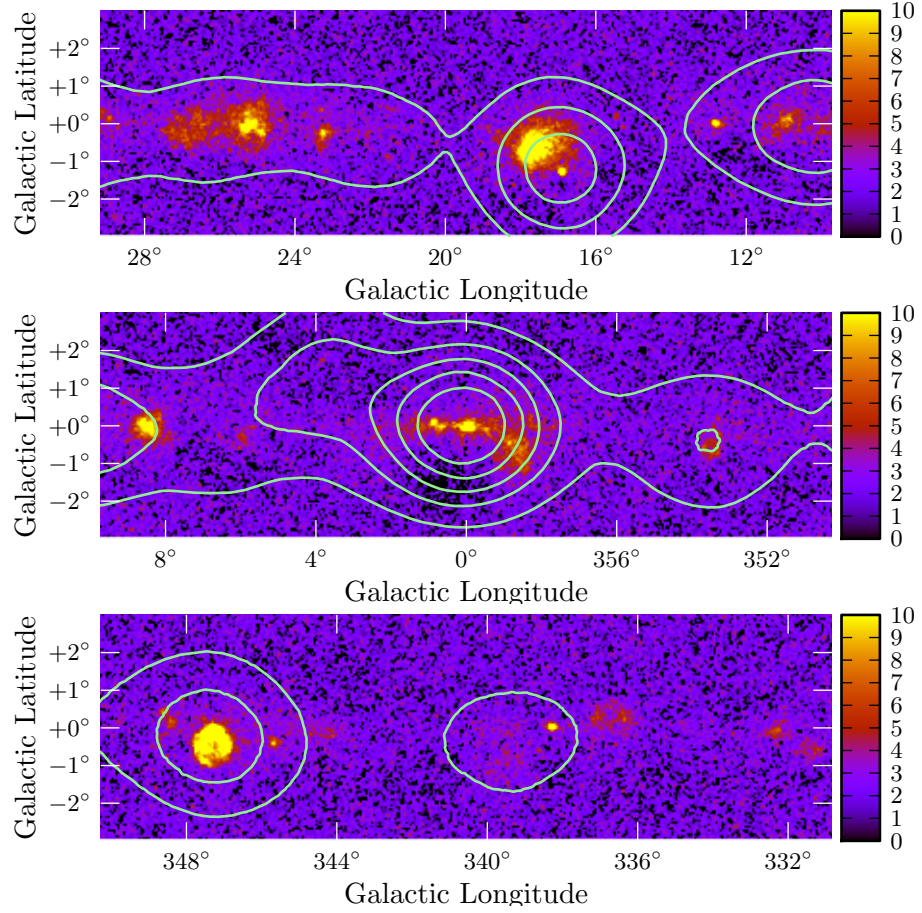


Figure 6.11: The strips show a Minkowski sky map of the inner part of the galactic plane. Color code, binning and analysis parameters correspond to those of figure 6.6, with the exception of the background level  $l$ , which is set to 0.5 for the galactic plane. The overlaid green contours show the corresponding acceptance map of the underlying data set in steps of 10%. The innermost ring around the galactic center corresponds to an acceptance level of 90%.

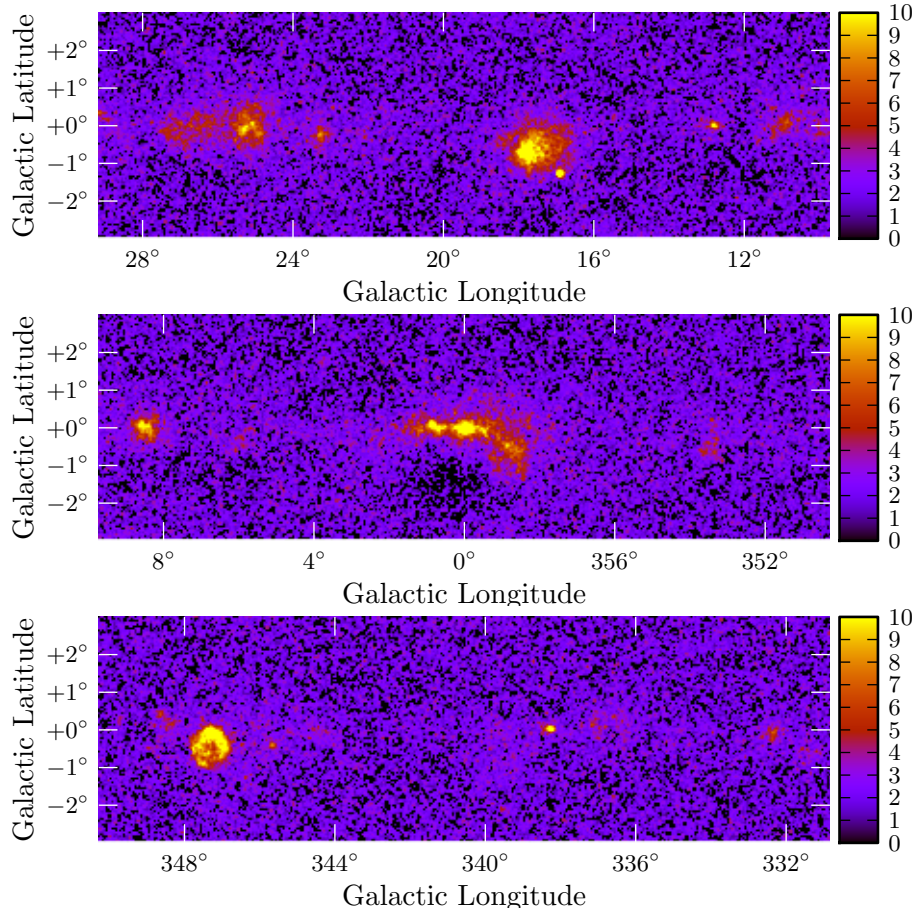


Figure 6.12: The strips show a Minkowski sky map of the inner part of the galactic plane. The shown region, color scale and all analysis parameters are identical to those of figure 6.11, except for the background level  $l$ , which is set to 1 here for comparison.

## 6.2 The Fermi Satellite Mission

The Fermi Gamma-ray Space Telescope (Fermi) mission is a satellite mission designed to study high energy gamma rays. Its primary instrument is the Large Area Telescope (LAT), an imaging gamma-ray telescope for the energy range from below 20 MeV to more than 300 GeV (Atwood et al., 2009). The LAT's large FoV<sup>6</sup> and its high sensitivity make it an excellent survey instrument. Therefore, the LAT is mostly operated in its all-sky scanning survey mode, which provides a quite uniform exposure of the entire sky seen from a low Earth orbit. The technical details of the LAT instrument are discussed in Atwood et al. (2009) and an overview of the analysis methods and recent results of the mission are given in Nolan et al. (2012) and references therein.

### 6.2.1 Background Model

Just like for H.E.S.S., the main source of background events for Fermi are charged cosmic rays. Nevertheless, unlike ground-based experiments, which need to deduce information about cosmic rays via secondary particles from air showers, satellite experiments directly measure the incident particle. This allows for a more efficient rejection of charged-particle background events than indirect approaches based on air showers. Thus, most of the charged particles are rejected already on the satellite and never make their way into the data sets used for analysis. Therefore, Fermi analyses are not affected very much by charged cosmic rays.

Still, Fermi analyses are affected by a large number of background events. However, these are due to diffuse gamma rays, which represent the dominant source of gamma rays in the energy range of Fermi. These diffuse gamma rays originate from secondary processes like interactions of very high energy cosmic rays and interstellar molecular clouds and form a background on top of which the primary sources of high energy gamma rays have to be identified. Based on differences in their spatial and spectral structure, the diffuse gamma rays can be differentiated into two main contributions: Extragalactic and galactic diffuse emission. While the extragalactic diffuse emission contributes a homogeneous and isotropic background throughout the entire sky, the galactic diffuse emission is governed by the mass density and distribution of local

---

<sup>6</sup>With a FoV of 2.4 sr the LAT observes nearly 20 % of the sky simultaneously.

molecular clouds in our galaxy. The details of the models used to describe these contributions are given in Nolan et al. (2012) and references therein.

The Fermi Science Support Center provides detailed documentation on the public software tools and input data needed to perform analyses of Fermi observations, as well as guides on how to obtain them. The tool *gtsrcmaps* described there can be used to compute model-predicted counts maps for complex models describing the entire gamma-ray sky, including diffuse background sources like the diffuse emission from the galactic plane as well as known primary sources of gamma rays. Applying this tool to a model consisting only of the extragalactic and galactic diffuse background sources you can compute the background model needed for a structure analysis. The resulting source maps contain all detector effects, like PSF, exposure and spatial sensitivity of the LAT. So, in the jargon of Fermi a H.E.S.S. acceptance map would be the source map of a homogeneous and isotropic diffuse background source. Using this analogy, the morphometric analysis of Fermi data is actually very similar to the process described in section 6.1.1, the only difference being that equation (6.1) has to be replaced by

$$\lambda_{ij} = \lambda_g g_{ij} + \lambda_e e_{ij}, \quad (6.5)$$

where  $\lambda_g$  and  $\lambda_e$  describe the background levels of the galactic and extragalactic diffuse emission, respectively, and  $g_{ij}$  and  $e_{ij}$  are the corresponding bins of the associated source maps. These background levels are determined as usual, using the robust fit introduced in section 5.3.

In principle this is enough to start analyzing Fermi data, if you trust the model-predicted counts. However, there are quite some uncertainties in the modeling of the diffuse gamma-ray background and thus it is useful to add some more flexibility. Instead of fitting only  $\lambda_g$  and  $\lambda_e$ , i.e. the flux level of the diffuse sources, it is reasonable to add some parameters describing the spectra of the background sources. As spectral modeling is beyond the scope of this work, the easiest solution is to perform the robust fit of  $\lambda_g$  and  $\lambda_e$  in energy slices. *gtsrcmaps* can be used to compute source maps binned in energy and the robust fit can be used to adjust  $\lambda_g$  and  $\lambda_e$  to each corresponding energy bin of the counts map. Fitting all energy bins independently gives great flexibility with respect to the spectral shape of the background sources, without the need to develop and fit a spectral model.

After the independent robust fits of all energy bins, each energy bin can be flattened to a certain background level using the methods described in chapter

5. The resulting background-corrected energy bins of the counts map can just be summed to produce the final energy-integrated background-corrected counts map. This is compatible with the background model, as the sum of Poisson-distributed counts with expectation values  $\mu_i$  is a Poisson-distributed count with expectation value  $\mu = \sum_i \mu_i$  and thus the Poisson structure of the counts is not destroyed by summing the different energy bins. So, the resulting counts map can be tested for deviations from the structure of a pure homogeneous and isotropic Poisson background and Minkowski maps can be used to visualize these deviations.

### 6.2.2 PKS 2155–304, a View of an Extragalactic Point Source

While PKS 2155–304 is a very prominent source for VHE gamma-ray telescopes like H.E.S.S., it is not very special in the energy range of the LAT. The primary reason to choose it as a first analysis target is its position. With a galactic latitude of  $-52.25^\circ$  it is far enough from the galactic plane to reduce the effect of uncertainties in the galactic diffuse model on the structure analysis. Furthermore, this region of the sky is quite empty compared to the number of sources which are seen by LAT in other regions of the sky. Thus PKS 2155–304 is a nice first target to study the general characteristics of the structure analysis of data taken with the LAT.

All LAT results presented here and in the following were produced using the Fermi Science Tools version v9r23p1 (Fermi Science Support Center), event selection criteria suggested in the Fermi Science Tools Cicerone (Fermi Science Support Center) with time cuts corresponding to the second LAT source catalog (Nolan et al., 2012) and the diffuse background models of this second catalog.

Figure 6.13 shows the resulting Minkowski map after using the robust fit to determine the background level in each energy bin and correcting for the background structure of the galactic and extragalactic diffuse background counts. The sky map shows some fainter sources in addition to the strong signal of PKS 2155–304 in the center. All of these sources have a counterpart in the second Fermi catalog, but as shown in figure 6.14 the official Fermi catalog comprises several sources in this region, which can not be seen in the Minkowski sky map. One of the reasons for this sensitivity difference is the spectral information, which is completely discarded in this simple first approach of a morphometric analysis. Summing all energy bins effectively

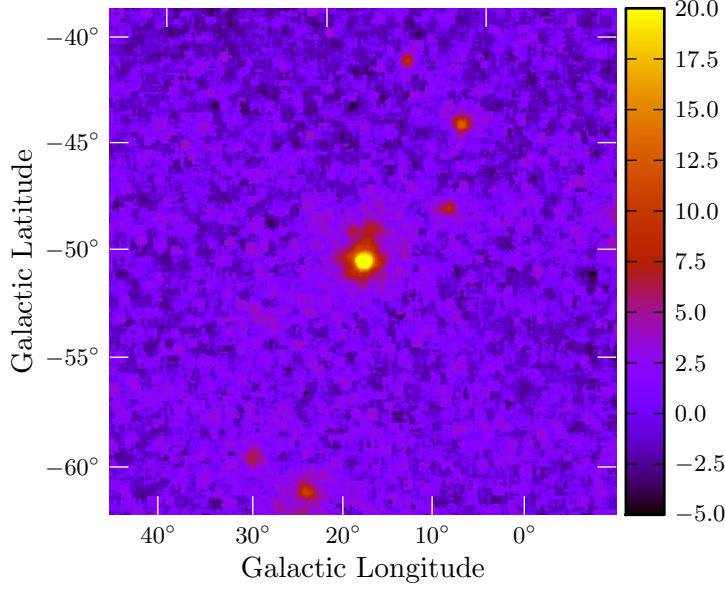


Figure 6.13: Minkowski sky map showing the local structure deviations in the region around PKS 2155–304 (at the center of the map). The color scale shows the significance of the local structure deviations around the corresponding bin in units of sign-augmented  $\sigma$  (cf. figure 6.6). The color scale is truncated to improve the visibility of regions around PKS 2155–304. The bin size of the sky map is  $0.1^\circ \times 0.1^\circ$ . The counts map used to compute this Minkowski map is based on the LAT events in the energy range from 100 MeV to 100 GeV and the time interval of the second Fermi catalog (Nolan et al., 2012).

reduces the analysis to an analysis of the low-energy counts. As the flux of gamma-ray sources drops rapidly with increasing energy, higher energy bins provide only a small contribution to the energy-integrated counts map compared to the contribution of low-energy bins.

A full likelihood analysis of the gamma-ray counts using spatial and spectral models as described in Mattox et al. (1996) uses the full information and can provide superior sensitivity. As the Fermi analysis is based on this likelihood



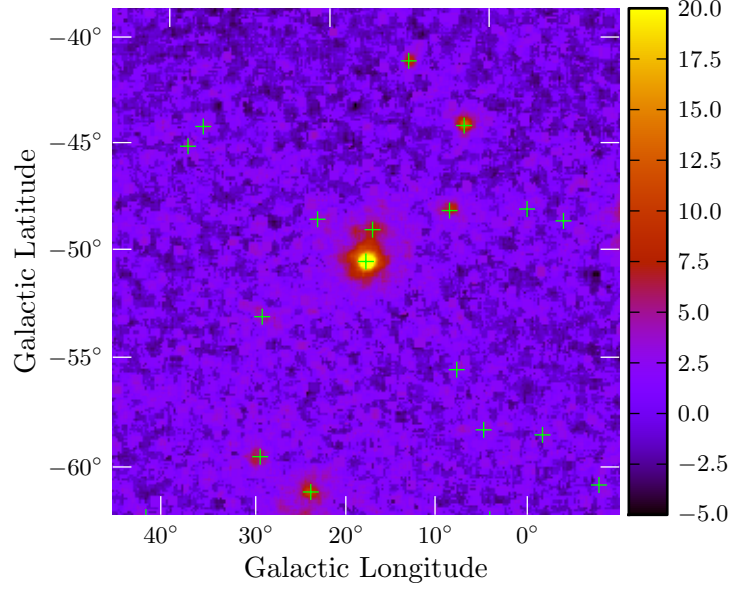


Figure 6.14: Same sky map as in figure 6.13 overlaid with the sources of the second Fermi catalog. The green crosses show the best fit positions of the Fermi catalog sources.

fit, it is not surprising that the Fermi analysis performs better in the region of the sky studied here. However, the big disadvantage of the likelihood analysis is the requirement to model all sources contributing to a counts map prior to the analysis of the counts map. This means that all contributing sources, including diffuse background sources, point sources with different spectral shapes and extended sources with different source morphologies, have to be modeled and included in the fit. Errors and uncertainties in these models lead to systematic errors in the likelihood analysis.

While the energy integrated structure analysis is currently not competitive compared to the standard Fermi likelihood analysis, it still provides a working (less sensitive) analysis framework for Fermi data. However, a major advantage of the structure analysis is its independence of source models (except for the background sources). So, without needing to model the source morphology and source spectrum prior to the analysis, it is still possible to find sources of



gamma rays as deviation from the expected background structure. Therefore, this complementary analysis approach may be used to check the likelihood analysis results for systematic errors which might arise from issues with the used source models.

### 6.2.3 RX J1713.7–3946, a View of a Galactic Source

Due to the intense background of diffuse gamma rays from the galactic plane it is very hard to detect and analyze sources in this region. Therefore, RX J1713.7–3946 has only recently been detected by Fermi (Abdo et al., 2011) and was not part of the bright source list (Abdo et al., 2009) or the first catalog (Abdo et al., 2010).

While RX J1713.7–3946 should in principle be visible using 24 months of LAT data (Abdo et al., 2011), it does also not appear on the Minkowski map shown in figure 6.15. This is due to uncertainties in the galactic diffuse model, which make it hard to detect the source on top of the pronounced background. Looking at figure 6.15 you find a dark cloud in the Minkowski map around the position of RX J1713.7–3946. These negative deviations indicate structural deviations from the expected background with less than the expected number of counts in the local scan window. Thus, the cloud corresponds to an extended region which does not agree well with the background model and where less than the model-predicted counts were detected. Presumably, this dark cloud in the Minkowski map corresponds to a region of the sky, where the galactic diffuse model overestimates the expected background. Due to this overestimation it is hard to find sources in this region (and incidentally there are no sources of the second Fermi catalog in the dark cloud), as additional counts from gamma-ray sources would make the data come closer to the model-predicted values, instead of introducing deviations from the background model.

Zooming in on the position of RX J1713.7–3946 shows that there is no significant deviation from the background model at the position of the H.E.S.S. source (cf. figure 6.16). So, how can the Fermi likelihood analysis detect the source in spite of the issues with the background model? The Fermi likelihood analysis uses a likelihood ratio test, similar to the one discussed in chapter 4. However, while the analysis introduced here does not restrict the alternative hypothesis, the Fermi likelihood analysis uses the background model with the additional source under test included in the model as only viable alternative

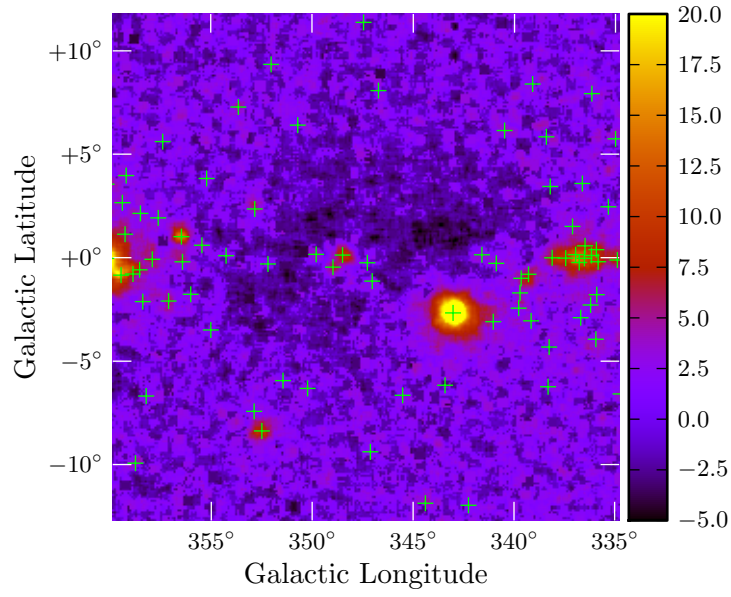


Figure 6.15: Minkowski sky map showing the local structure deviations in the region around RX J1713.7–3946 (in the center of the shown map). The color scale, image parameters and analysis parameters correspond to those of figure 6.13. The overlaid green crosses mark the best fit positions of the sources of the second official Fermi catalog (Nolan et al., 2012) in the shown region.

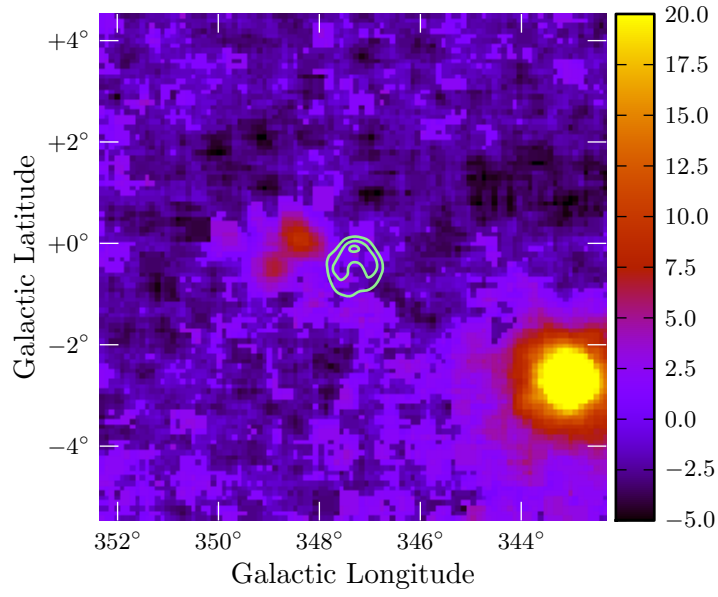


Figure 6.16: Sky map of figure 6.15 zoomed to the position of RX J1713.7–3946. The green contours show the significance contours of the smoothed Minkowski map from figure 6.8, i.e. the contours of the H.E.S.S. source.

to the background model excluding this source. Thus, the Fermi likelihood analysis kind of looks for improvements on the background model and may find a significant improvement even if both fits (the model with the additional source and the model without this source) are bad, as long as the additional source in the model improves the model significantly. The here introduced structure analysis on the other hand compares the background model not only to a model with one additional source, but to arbitrary models and thus gives an absolute estimate on how well the data agrees with the background model.

This relative estimation of the significance combined with the increased sensitivity due to the full spectral modeling of all sources allows the Fermi likelihood analysis to find significant improvements to source models, where the structure analysis can only point at the shortcomings of the used background model. So, the morphometric analysis needs a good understanding of the background in order to be able to find deviations from the expected background structure. Still, it does not need any models for the sources to be detected and it can be used to study the validity of the used background model.

## 7 Systematic Studies

The previous chapter shows some basic applications of the developed morphometric analysis framework with some promising results. This chapter characterizes the important features of the framework by quantifying relevant properties, which allows to compare the introduced framework to the standard tools available in gamma-ray astronomy.

### 7.1 Analysis Run-Time

As discussed in section 3.4.2, computing the distributions required for a structure analysis based on Minkowski tensors is quite time-consuming. Nevertheless, these distributions can be precomputed once for the desired combination of tensors and boundary conditions. Using these precomputed distributions to study gamma-ray counts maps is then amazingly fast. While the actual analysis run-time depends on the chosen set of tensors and the available hardware, the analysis of a  $100 \times 100$  bins counts map took  $\mathcal{O}(10\text{ s})$  for different sets of tensors on the TinyFat cluster (cf. section 3.4.2 for hardware details). For a given set of Minkowski tensors and a certain scan window size the run-time of an analysis of an  $N \times N$  bins counts map scales as  $N^2$ , since the number of bins determines how often the local structure of a scan window has to be evaluated. With the accessible scan window sizes of  $\mathcal{O}(5 \times 5)$  this leads to an expected run-time of  $\mathcal{O}(N^2 \cdot 10^{-3}\text{ s})$  for the analysis of an  $N \times N$  sky map.

Actually, when analyzing real gamma-ray data the main contribution to the overall run-time of the analysis is usually not caused by the local structure quantification but by the robust fit of the background model. Due to the simple model with only a few parameters compared to a full-blown model containing all sources, like in the standard Fermi analysis, this fit usually converges within  $\mathcal{O}(1\text{ min})$  for sky maps of  $100 \times 100$  bins.

All in all the run-time of the structure analysis is usually significantly shorter than the run-times of the standard H.E.S.S. or Fermi analyses. This makes the method interesting for tasks where prompt feedback is desired.

To mention but one possible application: The introduced structure analysis could be integrated into the data acquisition system of a telescope system like H.E.S.S. to give immediate feedback on newly taken data and to allow prompt follow-up observations if a significant structure fluctuation is detected in the observed sky.

## 7.2 Influence of Scan Window Size

The size of the scan window used to study the local structure of a gamma-ray counts map has a strong impact on the sensitivity of an analysis. Increasing the size of the scan window effectively increases the sensitive area of the local analysis and thus the sensitivity of the analysis.

Assuming that the number of expected background counts  $\lambda$  is known, the significance of a photon excess in a given bin can be approximated by

$$\sigma_p(k, \lambda) = \frac{k - \lambda}{\sqrt{\lambda}}, \quad (7.1)$$

where  $k$  is the number of detected gamma-like events in this bin. So,  $\sigma_p$  measures the deviation of the measured count  $k$  from the expected value  $\lambda$  in units of the standard deviation of a Poisson distribution with expectation value  $\lambda$ .<sup>1</sup> As the sum of Poisson-distributed values is Poisson-distributed, summing  $n$  bins with similar count values of about  $k$  and identical background levels  $\lambda$  leads to a significance of

$$\sigma_p(k, \lambda; n) \approx \frac{nk - n\lambda}{\sqrt{n\lambda}} = \sqrt{n} \frac{k - \lambda}{\sqrt{\lambda}} = \sqrt{n} \sigma_p(k, \lambda). \quad (7.2)$$

Thus, the significance of a gamma event excess in a region of  $N \times N$  bins is expected to scale like  $N$ . As the structure analysis is also sensitive to gamma-ray excess regions via its threshold dependent examination, it is plausible to

---

<sup>1</sup>The probability to measure a given  $\sigma_p$  in a Poisson distribution is *not* the same as measuring a deviation of  $\sigma = \sigma_p$  in a standard Gaussian distribution. Thus, you have to resort to methods like the likelihood ratio estimate of Li and Ma (1983) to find a normally distributed  $\sigma$ , which quantifies the significance level  $\alpha$  of a deviation (cf. table 4.1). Nevertheless,  $\sigma_p$  is a good approximation to the significance in units of  $\sigma$  and it allows to immediately see some features which are complicated to prove using the likelihood estimate. Therefore, this approximation is used in the following to motivate the expected behavior.

expect a similar scaling if there are no significant structures in the studied region.

The test pattern shown in figure 7.1 is used to verify the approximate predictions of the scaling properties. Using equation (7.1) this counts map can be transformed into an approximate significance map of local gamma-ray excess. The resulting maps for two different oversampling radii, i.e. integrating a different amount of bins around each location, are shown in figure 7.2. In order to verify the expected scaling property it is useful to examine the correlation of the pixel values of both maps. Figure 7.3 indeed shows the predicted linear dependency when increasing the size of the local

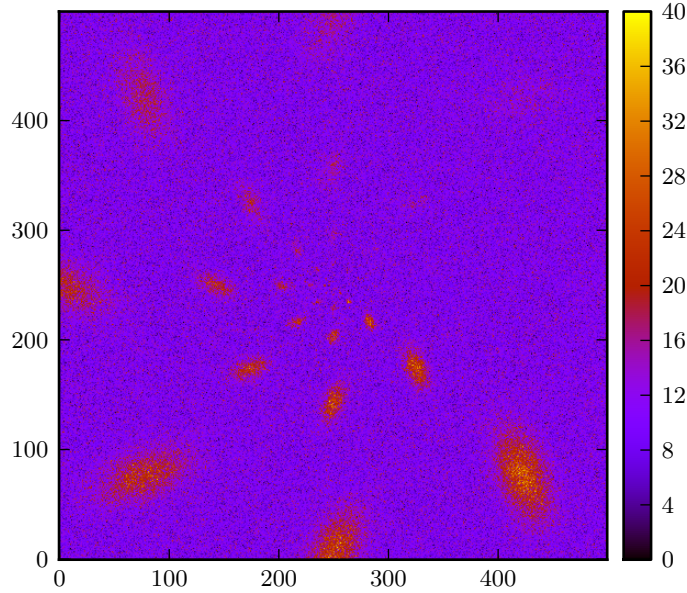


Figure 7.1:  $500 \times 500$  bins map of Monte Carlo simulated counts of a background level of  $\lambda = 10$  and various extended sources. The underlying model of the extended sources is a 2D Gaussian with an asymmetry of 2 (i.e.  $\sigma_x = 2\sigma_y$ , where  $\sigma_x^2$  and  $\sigma_y^2$  are the eigenvalues of the covariance matrix of the Gaussian). The counts map contains sources of different intensity and different length scale, to comprise a large set of different structures.

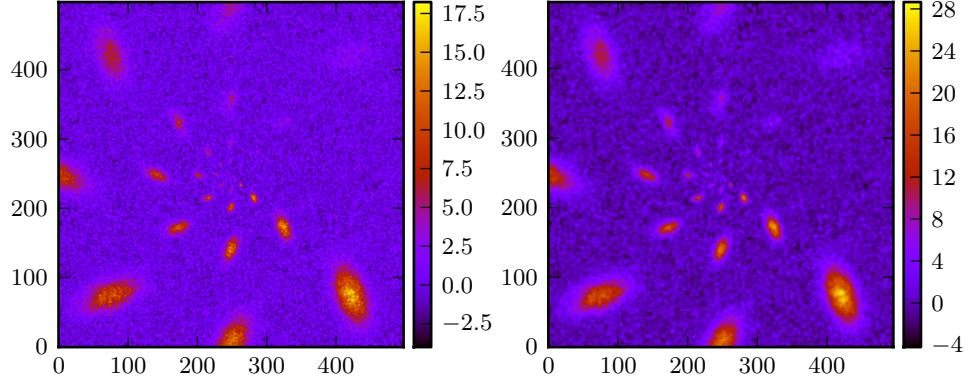


Figure 7.2: Significance maps of figure 7.1. Both maps use equation (7.1) to estimate the significance of the local gamma-ray excess. The left map sums the contribution of a  $3 \times 3$  bins region around each bin of the map in order to estimate the local excess, the right map uses a  $5 \times 5$  region. The color code shows the resulting estimate of the significance.

scan window. The pixel values of the  $5 \times 5$  bins scan window significance map are systematically larger than the corresponding values of the  $3 \times 3$  bins scan window map, which corresponds to the scatter points being on the right hand side of the green line. The fact that the scatter points show a nice linear dependency corresponds to the expected linear scaling of the significance. The width of this line of scatter points is due to statistical fluctuations of the summed bins, while equation (7.2) assumes the same number of counts for all summed bins.

The same technique can be used to study the scaling of the significance derived via the local structure analysis. Figure 7.4 shows the scatter plot of significance values obtained from Minkowski maps with different scan window



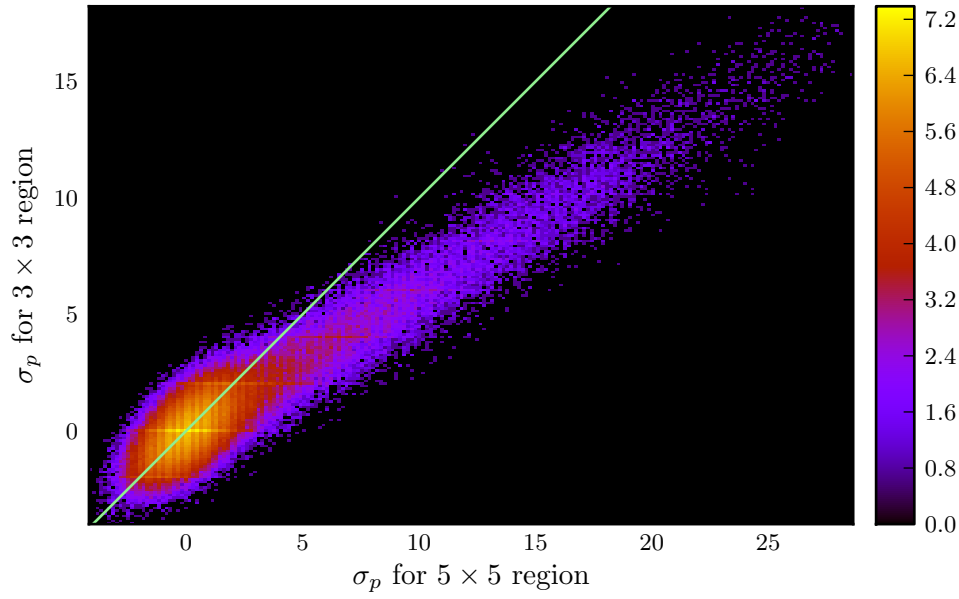


Figure 7.3: Pixel value correlation of the significance maps of figure 7.2. For each pixel of the significance maps the corresponding values of the two maps are used as coordinates for a new point in a scatter plot. The density of these scatter points is visualized by the color code in arbitrary units. The green line shows the identity function, which corresponds to a correlation coefficient of 1.

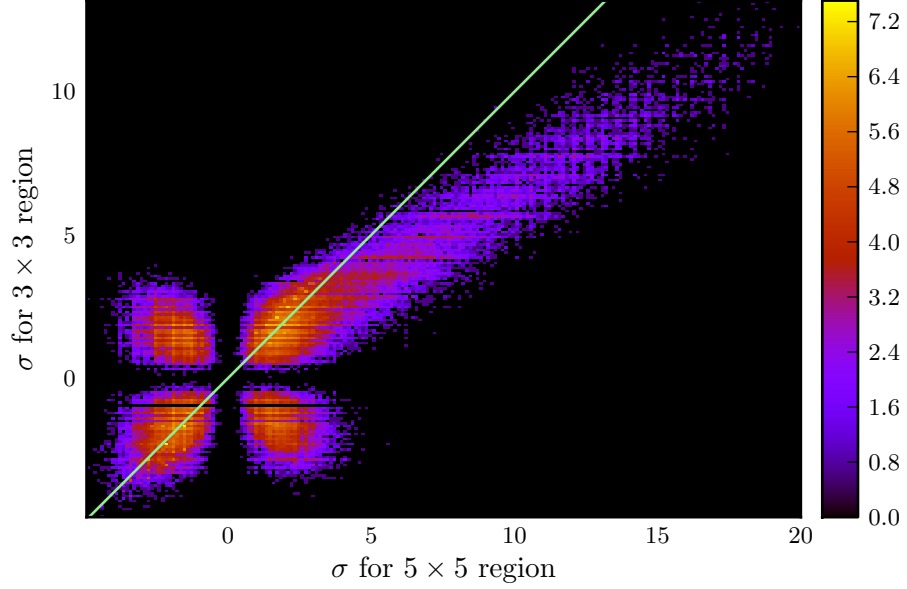


Figure 7.4: Pixel value correlation of Minkowski maps computed from the counts map of figure 7.1 with different scan window sizes. The corresponding scan windows used to study the local structure deviations are  $3 \times 3$  and  $5 \times 5$  bins in size respectively. Color scale and green line are analogous to figure 7.3.

sizes. The main message is very similar to that of figure 7.3: A larger scan window leads to an increase of the significance and the linear alignment of the scatter points supports the proposed linear scaling. While this answers the question of the influence of the scan window size, some of the features visible in the scatter plot deserve further discussion.

The empty strips at  $\sigma = 0$  are the result of choosing the maximum deviation strength to characterize the local structure. For a pixel of a Minkowski map to be 0 the structure of all b/w images for all thresholds has to agree perfectly with the model prediction, which is highly unlikely in the light of statistical fluctuations.

Another interesting feature are the regions of negative correlation, i.e. the regions perpendicular to the green line, where pixels have negative deviation

strength in one Minkowski map but positive in the other one. Figure 7.3 does not show these extended regions of negative correlation, as it shows the significance of the local excess and except for statistical fluctuations the different significance maps should agree on the point whether there are too many or too few events in the local scan window. Figure 7.4 on the other hand does not show the significance of an excess, but a structural deviation from Poisson noise. The sign of a pixel in a Minkowski map is chosen with respect to the number of black pixels at the selected threshold compared to the number of expected pixels at this threshold. If the thresholds were equal in both maps, regardless of the scan window size, you could expect the same sign and no negative correlation, except for statistical fluctuations, but the thresholds do not have to be equal. As the threshold is chosen according to the strongest deviation, it may well lie at a completely different level using a larger scan window if the local region shows only background noise<sup>2</sup> and the expected number of black pixels may be very different and thus may lead to a different sign.

Summarizing, the values of significance maps based on the excess counts of a sky map as well as the significance of Minkowski sky maps scales linearly with the size of the used scan window. Therefore, when comparing sensitivities of different analysis methods you always need to make sure that identical scan window sizes are used or that this effect is properly corrected for.

### 7.3 Influence of Background Correction

The background correction introduced in chapter 5 reduces the sensitivity of an analysis to gamma-ray sources on top of this corrected background. Just like in the previous section, equation (7.1) provides a good basis to estimate this effect.

Let  $\lambda$  be the expected number of counts in a bin, as predicted by the background model and let  $\lambda_f$  be the target background level after background correction, then the relative change of the background level is given by  $\epsilon = \lambda_f/\lambda$ .

---

<sup>2</sup>If the scan window contains only random fluctuations compatible with the background, then all b/w pixel images at all thresholds are compatible with the model prediction and their deviation strength is similar. Thus, the threshold of maximal deviation strength is randomly distributed among the possible values. Adding new bins with new statistical fluctuations by widening the scan window restarts this random process of selecting a threshold and may result in completely different values.

If the counts  $k$  in a bin are reduced via post-selection, then  $\epsilon k$  counts are expected to remain after this background correction. Thus, after post-selection the significance of an excess is approximately given by

$$\frac{\epsilon k - \epsilon \lambda}{\sqrt{\epsilon \lambda}} = \sqrt{\epsilon} \sigma_p(k, \lambda). \quad (7.3)$$

If the bin is filled with Monte Carlo events during the background correction, then  $k + (\lambda_f - \lambda)$  counts are expected in the corrected bin. Therefore, the significance of the Poisson filled bin is approximately given by

$$\frac{k + (\lambda_f - \lambda) - \lambda_f}{\sqrt{\epsilon \lambda}} = \frac{1}{\sqrt{\epsilon}} \sigma_p(k, \lambda). \quad (7.4)$$

Thus, the significance of a background-corrected bin is in general approximately given by

$$\sigma_p(k, \lambda; \epsilon) = \begin{cases} \sqrt{\epsilon} \sigma_p(k, \lambda) & \forall \epsilon \leq 1 \\ \frac{1}{\sqrt{\epsilon}} \sigma_p(k, \lambda) & \forall \epsilon > 1. \end{cases} \quad (7.5)$$

With  $\epsilon = 1 + \delta$  this yields

$$\sigma_p(k, \lambda; \epsilon) = \begin{cases} \left(1 + \frac{\delta}{2}\right) \sigma_p(k, \lambda) + \mathcal{O}(\delta^2) & \forall \epsilon \leq 1 \\ \left(1 - \frac{\delta}{2}\right) \sigma_p(k, \lambda) + \mathcal{O}(\delta^2) & \forall \epsilon > 1. \end{cases} \quad (7.6)$$

Thus, for a relative change of the background level of  $|\delta|$  the corresponding significance decreases by a fraction of  $|\delta|/2$  for small  $|\delta|$ .

The general dependency of  $\sigma_p$  on the relative change of the background level  $\epsilon$  is shown in figure 7.5. So, while covering the data with a layer of Monte Carlo events may seem very intrusive at first, it actually has a smaller impact on the significance of excess counts for large background corrections than post-selection.

## 7.4 Sensitivity of Different Tensors

After examining some more general features in the previous sections, this section finally shows the effects of using different Minkowski tensors. However,

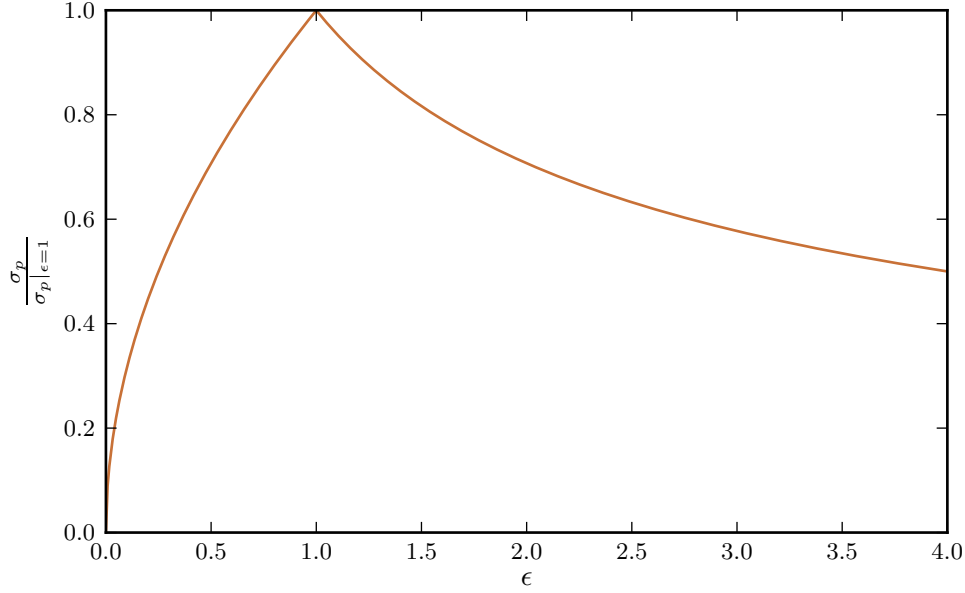


Figure 7.5: Relative change of the significance  $\sigma_p$  with respect to changing the background level by a factor of  $\epsilon$  using the background correction discussed in chapter 5.

because of the unknown trial factors caused by choosing the threshold of maximal deviation strength, a direct comparison of the results of the structure analysis to a significance map based on the likelihood ratio test of Li and Ma (1983) is meaningless. Nevertheless, it is possible to compare different combinations of tensors to a structure analysis using only the area functional. The analysis based exclusively on the area functional is the structure analysis equivalent of an analysis based on excess counts, as it only evaluates the number of bins above a given threshold.

As the result of a local structure analysis depends on many factors, like the intensity and length scale of present structures, the easiest way to get an impression of the general features is following the ideas of section 7.2. Using the counts map of figure 7.1 as input to the local structure analysis covers a large range of different intensities and length scales and allows to study the tendencies in the distributions of Minkowski map pixel values.

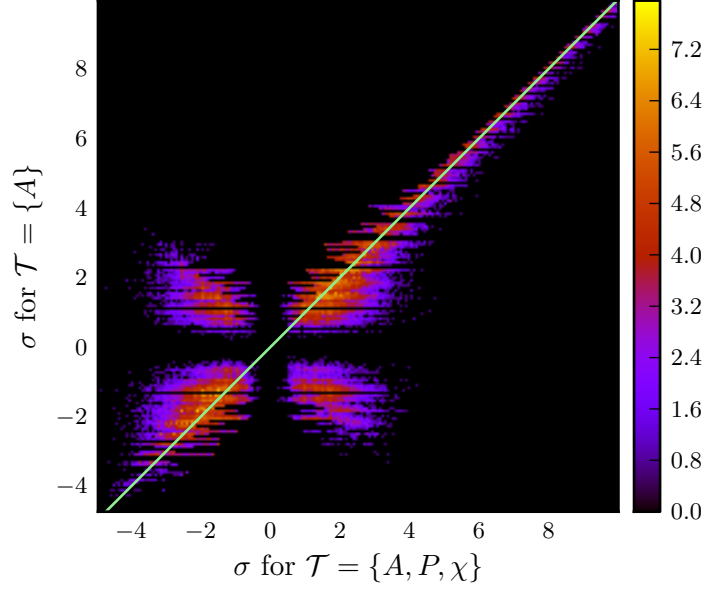


Figure 7.6: Scatter plot of the significance values of two Minkowski maps of the counts map of figure 7.1 (cf. figure 7.4 for the general concept of such scatter plots). The  $x$ -axis shows the significance of a Minkowski map based on the Minkowski functionals area  $A$ , perimeter  $P$  and Euler characteristic  $\chi$ , the  $y$ -axis shows the significance of a Minkowski map based on  $A$  only.

Figure 7.6 shows a scatter plot comparing the pixel values of two Minkowski maps using different sets of Minkowski functionals. A systematic shift to larger significance for the Minkowski map based on the combination of several Minkowski functionals compared to the map based on the area only can be seen. This shift is not a systematic error in the performed analysis. The bulk of the pixels of both Minkowski maps is in very good agreement (red regions of the scatter plot) and only a few pixels tend to larger values. These pixels correspond to regions where the local structure of the counts map exhibits special features which are detected by the additional Minkowski functionals and lead to the increased significance. This is clearly visible in figure 7.7, where the ratio of the two Minkowski maps is shown. The pixels

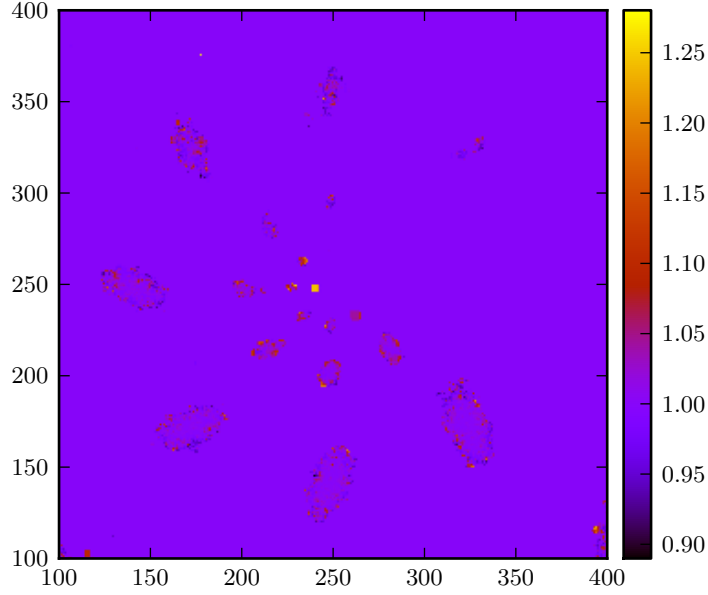


Figure 7.7: This map shows the inner region of the ratio of the two Minkowski maps used for figure 7.6. To suppress the large fluctuations in regions compatible with the background model, the ratio was set to 1 for all bins below  $4\sigma$  in one of the original Minkowski maps.

of enhanced significance are located at sources with a length scale similar to the size of the scan window and they coincide with the edges and curvatures of these sources. This perfectly fits the profile of the additional functionals, as the perimeter and the Euler characteristic are sensitive to edges and local curvatures, respectively.

The observed increase of about 10% in significance does not seem too impressive, but might already help to lift weak extended sources over the detection threshold. Larger scan windows, containing more quantifiable structure, probably help to improve the sensitivity to extended sources even further.

Figure 7.8 shows a similar sensitivity comparison between the area functional and the area combined with its tensor of rank 1. Even this one additional low rank tensor causes a sensitivity gain similar to the combined use of the 3

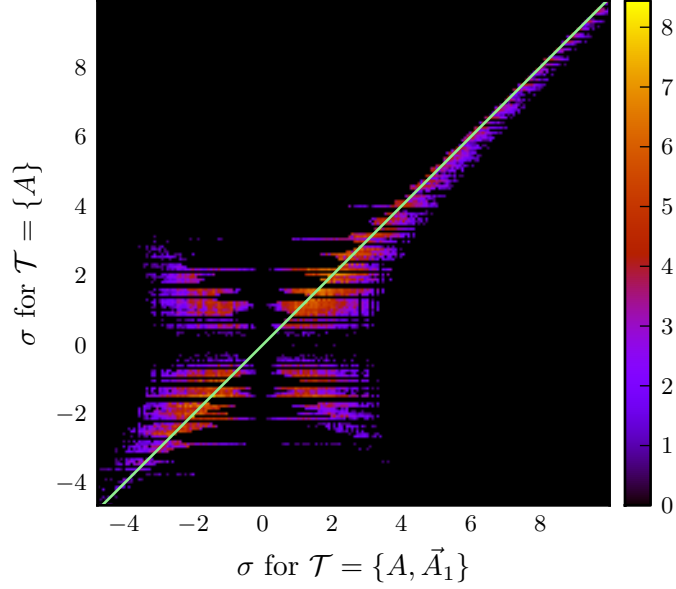


Figure 7.8: Sensitivity comparison of Minkowski maps based on different sets of Minkowski tensors analogous to figure 7.6. The  $x$ -axis shows the significance values of a Minkowski map based on the area  $A$  and its rank 1 tensor  $\vec{A}_1$ , the  $y$ -axis the values of a Minkowski map based on the area only.

Minkowski functionals. The area tensor  $\vec{A}_1$  is proportional to the center of mass vector (cf. section 2.2) and if that vector does not point to the center of the scan window, this indicates an asymmetry in the observed counts, like e.g. a gradient within the scan window. As the edges of sources cause such gradients within the scan window, the additional tensor improves the sensitivity at the edges of sources. In fact, the ratio of the two Minkowski maps looks very similar to figure 7.7 and is therefore not shown here.

While it is reasonable to assume that the sensitivity can be increased further by combining more functionals, this approach is actually limited by the available scan window size. If you use enough Minkowski tensors to be able to uniquely identify every possible scan window configuration, the sensitivity is just as bad as when using the area functional only. This is due



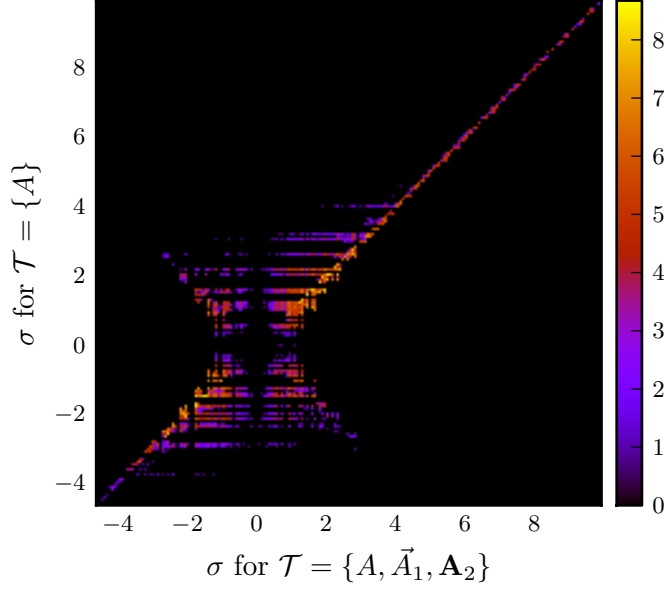


Figure 7.9: Sensitivity comparison of Minkowski maps based on different sets of Minkowski tensors analogous to figure 7.6. The  $x$ -axis shows the significance values of a Minkowski map based on the area  $A$  and its rank 1 and 2 tensors  $\vec{A}_1$  and  $\mathbf{A}_2$ , the  $y$ -axis the values of a Minkowski map based on the area only.

to the underlying homogeneous and isotropic background model, which makes every configuration with an equal count of black and white pixels equally probable. If you use enough tensors to discriminate between all of them you end up computing the contribution of  $A$  only, when integrating the critical region to determine the deviation strength. Only a limited set of tensors allows to ignore just enough information to find similarities among different configurations and to group the configurations into classes of typical and less typical structures. Using too many tensors, all of these classes contain exactly one element, which makes all of them equally typical. The effect of using too many tensors is shown in figure 7.9, where the area functional is compared to the set of tensors  $\mathcal{T} = \{A, \vec{A}_1, \mathbf{A}_2\}$ , where  $\mathbf{A}_2$  is the area tensor of rank 2. The shown correlation is very strong and there is virtually no difference

between the two Minkowski maps at deviation strengths beyond  $5\sigma$ .

Summarizing, the sensitivity of a counts map analysis can be increased by using Minkowski tensors. Currently the observable gain is only limited, but larger scan windows and the combination of additional Minkowski tensors may well lead to a powerful boost in future analyses.

## 7.5 Influence of Boundary Conditions

As discussed in chapter 2, the choice of boundary conditions may have a severe impact on the value of Minkowski tensors and even on their transformation properties. Therefore, it is important to study the influence of different boundary conditions on the introduced analysis framework.

Figure 7.10 shows a scatter plot comparing Minkowski maps using different boundary conditions. Fortunately, the shown distribution is perfectly symmetric with respect to the identity function (green line) and thus there are no systematic differences between the two Minkowski maps. So, while the values of the Minkowski tensors may be different using different boundary conditions, the probability distributions are constructed based on the values of the Minkowski tensors including their boundary contributions and thus each distribution accounts for these boundary conditions. Therefore, when determining the significance of a local structure value, the corresponding probability distribution helps to distinguish between typical and exceptional values independently of the absolute value of the structure functional.

The width of the distribution shown in the scatter plot is *not* due to statistical fluctuations, as both analyses use the same input counts map and the same scan window size, i.e. exactly the same pixel values in each examination of the local structure. Nevertheless, different probability distributions for the different boundary conditions lead to slightly different critical regions and thus to slightly different deviation strength (cf. chapter 4). This is perfectly fine and does not imply that one of the analyses is wrong and the other one yields the only true value quantifying the significance of the source detection. Actually the deviation strength does not quantify the significance of a detection at all (and neither does the likelihood ratio test of Li and Ma (1983)), it is a measure of the probability of errors of the first kind, i.e. of rejecting the background model while it is actually true (just like the likelihood ratio test of Li and Ma (1983)). Furthermore, this estimation of

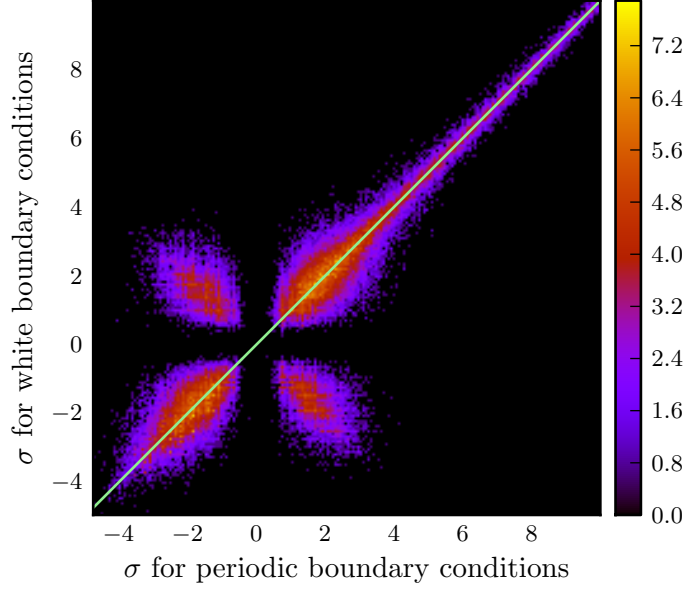


Figure 7.10: Scatter plot similar to 7.6, comparing Minkowski maps based on different boundary conditions of the local scan window. The  $x$ -axis shows the significance of a Minkowski map based on periodic boundary conditions, the  $y$ -axis shows the significance of a map based on setting the exterior of the scan window to white.

the probability of an error of the first kind does only hold if you always use the same criterion for rejecting the background model. As these criteria (i.e. the critical regions) are slightly different for different probability distributions of Minkowski tensors with different boundary conditions, some configurations may lie within a critical region of a certain significance level using one of the boundary conditions, while they may lie slightly beyond the critical region using different boundary conditions. The symmetry of the scatter plot demonstrates that this effect does not introduce a general bias in the quantification of structure deviations and that both systems can be used to determine a self-consistent deviation strength which is compatible with the deviation strength determined using different boundary conditions.

Some care has to be taken when using boundary conditions like minus

sampling, though. As minus sampling boundary conditions use a part of the information of surrounding bins to fix their boundary, they effectively increase their sensitive area compared to an  $N \times N$  scan window with boundary conditions that strictly confine all contributions to the interior of the scan window. Minus sampling does not use all the information of the surrounding bins and thus the sensitivity of the resulting scan window is smaller than the sensitivity of a  $(1 + N + 1) \times (1 + N + 1)$  window, where all adjacent lines of bins are fully included, but still the sensitivity is larger than the usual  $N \times N$  sensitivity due to the additional information at the boundaries. As the complexity to compute the distribution of a  $N \times N$  scan window with minus sampling boundary conditions is equivalent to computing the distribution of a  $(1 + N + 1) \times (1 + N + 1)$  scan window of other boundary conditions, while at the same time the sensitivity is lower, it is usually advisable to use different boundary conditions, like setting the exterior of the scan window to white.

## 7.6 Influence of Bin Size

The binning of the underlying counts map is a very influential parameter for structure analysis. You can e.g. hide even a rich structure like the one of RX J1713.7–3946 by choosing a bin size larger than the source itself. The resulting counts map may contain a bin with a significant excess compared to the expected number of background counts, but there is no inherent structure of the source left.

While too coarse binning always causes a loss of structure information, the impact of a very fine binning depends on the selected set of Minkowski tensors. Suppose you choose a binning where every gamma-like event resides in its own bin, separated by many empty bins from the next occupied bin. Using only the Minkowski functionals area, perimeter and Euler characteristic, there is no quantifiable structure in such a setup except for the number of occupied bins.<sup>3</sup> So, for these functionals this setup is equivalent to a very coarse binning, where all events fall into the same bin and the only quantifiable structure is the number of counts in this single bin. Using coordinate dependent Minkowski

---

<sup>3</sup>As all the occupied bins are disjoint, they contribute to the final structure functionals independently. Each bin counts for an area of 1, a perimeter of 4 and an Euler characteristic of 1. Thus, the final structure functionals are all proportional to the number of occupied bins.

tensors, on the other hand, it is possible to test the spatial distribution of the disjoint bins for homogeneity and isotropy. This dependency on the selected set of Minkowski tensors makes it hard to find general rules for an efficient choice of the bin size. Nevertheless, there are some guidelines which can help to find a solid setup.

The smallest structures detectable in the data of a telescope are determined by its PSF. The bin size of a counts map should not be larger than the typical size of this PSF, as otherwise the inherent structures of the data are smeared out. On the other hand, the size of the scan window used to study the local structure of the counts map should at least be as large as the PSF, as studying the local structure on smaller scales is futile. With the currently available scan window sizes of  $\mathcal{O}(5 \times 5)$  bins, these two constraints (bin size not larger than PSF and size of 5 bins larger than PSF) narrow the available bin sizes down quite considerably. The answer to the question which end of this range to choose (large bins with a size comparable to the PSF and therefore also a large scan window, or a scan window size comparable to the size of the PSF and therefore small bins) depends on the use case. If it is important to reduce binning effects, e.g. if point sources need to be approximately circular, you need to go with the small bin sizes. If the details of small structures are a minor concern, the larger bin size with the corresponding larger scan window allows to study larger structures. Thus, if you want to study large scale structures, you should choose the large scan window, if you want to study small structures in detail, you should choose the smaller bins.

## 8 Conclusion

### 8.1 Summary

The developed framework for the analysis of counts maps based on local deviations of their structure from the typical structure of a background measurement works very well. After computing the needed lookups, i.e. the probability distributions of the chosen Minkowski tensors, the analysis is as fast and simple to use as the likelihood ratio method suggested by Li and Ma (1983), but it leverages additional structure information to increase the sensitivity compared to an estimation of the local excess counts.

In contrast to using e.g. a full likelihood model fit to the data as done, for instance, by the Fermi Science Tools, you do not need to model potential sources before starting the analysis. You just need to model the expected background and sources can be detected by looking for significant deviations from this background model. This reduces the number of free model parameters significantly and makes the analysis run amazingly fast. You do not even need to exclude regions of known gamma-ray sources from the fit determining the background level, as the introduced robust fit automatically selects appropriate exclusion regions, as long as more than 50 % of the input data provide a reasonable estimate of the background.

The developed framework is very general and has already been applied successfully to data of the H.E.S.S. experiment and data of the Fermi LAT. It should be fairly easy to adopt the framework to further applications and experiments. In principle the framework could be used in many different kinds of applications where the structure of counts needs to be analyzed, and is not limited to gamma-ray astronomy.

The sensitivity gain observed so far is moderate. Nevertheless, the next generation of high performance computers will open the door to larger scan windows, which might increase the sensitivity gain considerably. There is just too little structure in a  $5 \times 5$  or even  $7 \times 7$  b/w pixel image (just imagine how many different pictures you could paint on a canvas of this size); on the other

hand, already  $16 \times 16$  pixel images allow a considerable range of different structures (this is the most common size of favicons,<sup>1</sup> which are used to show numerous different logos and icons). Therefore, it is reasonable to assume that considerable improvements can be expected as soon as it is possible to compute the probability distributions for scan windows of about twice the linear size of the windows available so far. This should soon be possible using future powerful computers.

Besides sensitivity gains for the analysis of gamma-ray counts maps, the introduced framework contains two aspects which are very interesting in their own right. The algorithm provided in chapter 3 for computing distributions of Minkowski tensors is the first algorithm ever which allows to compute these distributions for arbitrary Minkowski tensors with a run-time complexity better than the well known brute force alternative. Its generality and performance make it an interesting tool for any application making use of Minkowski tensors to study binned data. The second interesting aspect is the robust fit introduced in section 5.3, which was developed to adjust background models to data of the H.E.S.S. telescopes or the Fermi LAT without the need to model known gamma-ray sources or to maintain a list of exclusion regions describing the size of these sources. As selecting a good set of exclusion regions is one of the most delicate and error prone steps in the standard H.E.S.S. analysis, the presented robust fit could be used to automate this process and eliminate this source of error from the analysis.

## 8.2 Outlook

As already mentioned in the previous section, computing probability distributions of Minkowski tensors on future computers will open the door to larger scan windows. Already doubling the linear size of these windows compared to the currently available size is expected to increase the amount of quantifiable structures in the scan window considerably and thus the sensitivity gain is expected to increase substantially. This is very promising in the light of the start of H.E.S.S. Phase II with the upcoming inauguration of a new fifth Cherenkov telescope at the H.E.S.S. site. This telescope is supposed to lower the energy threshold of the experiment and — as known from the results of

---

<sup>1</sup>Favicons are icons shown in front of the URL in the address field of a web browser or in front of browser bookmarks.

the Fermi LAT — there are many extended structures visible at these energy scales, like the diffuse emission from the galactic plane. Future experiments like CTA with higher angular resolution and sensitivity promise to unveil even more structures in the gamma-ray sky, e.g. by resolving the morphology of extended sources which are point-like within the limits of the angular resolution of H.E.S.S. This increasing amount of quantifiable structures makes the introduced analysis framework a promising alternative to established analyses.

In addition to applying and refining the developed analysis framework, it would also be very interesting to study complementary structure analysis approaches. One could e.g. try to build an unbinned analysis of the spatial structure of gamma-like events by painting a square into a sky map for each event using a side length of  $a$ , aligning the square with the coordinate grid and centering it at the position of the event. By varying the parameter  $a$  the structure of the drawn area changes from disjoint squares to larger connected regions. A small source on top of otherwise homogeneously distributed background events results in a clustering of events at the source position, which makes the squares at this position intersect at smaller  $a$  than in other regions. Studying the structure of “event squares” as a function of  $a$  could therefore be used to identify sources on top of a homogeneous and isotropic background. As the intersections of aligned squares are always rectangles, the additivity of the Minkowski tensors could be used to break the structure quantification down to sums of rectangles, which makes the structure quantification very efficient. Similar approaches — usually using discs instead of squares — are very common in statistical physics (e.g. Mecke, 2000) and might provide interesting insights into the spatial structure of detected gamma-like events.

Thus, while the first steps are taken and a working structure analysis for gamma-ray counts maps based on Minkowski tensors has been established, there are still a lot of interesting questions ahead. Studying the presented analysis framework with larger scan windows and exploring alternative approaches like the method of “event squares” are just two of them. Considering future experiments with increased sensitivity as well as the steadily increasing exposure of current experiments like H.E.S.S., the structures observable in gamma-ray astronomy data will increase continuously and an analysis taking them into account provides a better understanding of the observed events.



# List of Figures

1.1	H.E.S.S. site . . . . .	11
1.2	H.E.S.S. Galactic Survey . . . . .	12
1.3	Fermi Bubbles . . . . .	13
2.1	Pixel Coordinates and Order . . . . .	25
2.2	Periodic Boundary Conditions . . . . .	31
2.3	Indistinguishable Configurations . . . . .	33
2.4	Problems of Periodic Boundary Conditions . . . . .	44
2.5	Marching Square Configurations . . . . .	46
2.6	Marching Square Example . . . . .	47
2.7	Triangular Pixels . . . . .	47
3.1	Contribution of Interior Pixels to Distribution . . . . .	58
3.2	Minus Sampling Contribution of First Corner Pixel . . . . .	60
3.3	Minus Sampling Contribution of First Row . . . . .	61
3.4	Minus Sampling Contribution of Last Pixel in First Row . . . . .	62
3.5	Minus Sampling Contribution of First Pixel in Row . . . . .	63
3.6	Minus Sampling Contribution of Last Pixel in Row . . . . .	64
3.7	Minus Sampling Contribution of First Pixel in Last Row . . . . .	65
3.8	Minus Sampling Contribution of Last Row . . . . .	66
3.9	Minus Sampling Contribution of Last Corner Pixel . . . . .	67
3.10	Periodic Contribution of First Corner Pixel . . . . .	69
3.11	Periodic Contribution of First Row . . . . .	70
3.12	Periodic Contribution of Last Pixel in First Row . . . . .	71
3.13	Periodic Contribution of First Pixel in Row . . . . .	71
3.14	Periodic Contribution of Last Pixel in Row . . . . .	72
3.15	Periodic Contribution of First Pixel in Last Row . . . . .	72
3.16	Periodic Contribution of Last Row . . . . .	73
3.17	Periodic Contribution of Last Corner Pixel . . . . .	73

## List of Figures

---

3.18	Algorithm Run-Time . . . . .	80
3.19	Algorithm Memory Usage . . . . .	82
4.1	Background Counts Map . . . . .	91
4.2	Local Structure Deviations . . . . .	92
4.3	Contribution to Minkowski Map . . . . .	93
4.4	Minkowski Map . . . . .	94
4.5	Sign-Augmented Minkowski Map . . . . .	96
5.1	Influence of Outliers on Different Fit Algorithms . . . . .	102
5.2	Intermediate Steps of Robust Fit . . . . .	105
5.3	Function to Minimize for Robust Fit . . . . .	107
6.1	Counts Map of the Crab Nebula . . . . .	114
6.2	Counts Map of the Surroundings of the Crab Nebula . . . . .	114
6.3	Acceptance Map of the Crab Nebula Observations . . . . .	115
6.4	Acceptance-Corrected Counts Map of the Crab Nebula . . . . .	116
6.5	Contribution of Acceptance Correction . . . . .	117
6.6	Minkowski Map of the Crab Nebula . . . . .	118
6.7	Significance Map of the Crab Nebula . . . . .	119
6.8	Minkowski Map of RX J1713.7–3946 . . . . .	121
6.9	Significance Map of RX J1713.7–3946 . . . . .	122
6.10	High Significance Maps of RX J1713.7–3946 . . . . .	123
6.11	Minkowski Map of the inner Galactic Plane . . . . .	126
6.12	High Background Minkowski Map of the inner Galactic Plane . . . . .	127
6.13	Minkowski Map of PKS 2155–304 . . . . .	131
6.14	Fermi Catalog around PKS 2155–304 . . . . .	132
6.15	Minkowski Map of RX J1713.7–3946 Fermi Data . . . . .	134
6.16	Fermi Data and H.E.S.S. Contours of RX J1713.7–3946 . . . . .	135
7.1	Test Pattern . . . . .	139
7.2	Excess Significance . . . . .	140
7.3	Excess Significance Correlation . . . . .	141
7.4	Minkowski Significance Correlation . . . . .	142
7.5	Impact of Background Correction . . . . .	145
7.6	$A$ vs $A, P, \chi$ . . . . .	146
7.7	Location of Sensitivity Changes . . . . .	147

7.8	$A$ vs $A, \vec{A}_1$ . . . . .	148
7.9	$A$ vs $A, \vec{A}_1, \mathbf{A}_2$ . . . . .	149
7.10	Impact of Boundary Conditions . . . . .	151

List of Tables

2.1 Minkowski Functionals of Image Components . . . . . 24

2.2 Pixel Contribution to Minkowski Functionals . . . . . 26

2.3 Symmetric Contributions to Minkowski Functionals . . . . . 27

2.4 Minus Sampling Contributions to Minkowski Functionals . . 29

2.5 Contributions to the Area Tensors . . . . . 41

2.6 Contributions to the Perimeter Tensors . . . . . 42

2.7 Contributions to the Curvature Tensors . . . . . 43

3.1 Mapping of Color Transitions to Operator Formulas . . . . . 69

4.1 Significances in different Units . . . . . 88

## Bibliography

- A. A. Abdo et al. Fermi/Large Area Telescope Bright Gamma-Ray Source List. *The Astrophysical Journal Supplement Series*, 183(1):46, 2009. doi: 10.1088/0067-0049/183/1/46.
- A. A. Abdo et al. Fermi Large Area Telescope First Source Catalog. *The Astrophysical Journal Supplement Series*, 188(2):405, 2010. doi: 10.1088/0067-0049/188/2/405.
- A. A. Abdo et al. Observations of the Young Supernova Remnant RX J1713.7–3946 with the Fermi Large Area Telescope. *The Astrophysical Journal*, 734(1):28, 2011. doi: 10.1088/0004-637X/734/1/28.
- M. Actis et al. Design concepts for the Cherenkov Telescope Array CTA: an advanced facility for ground-based high-energy gamma-ray astronomy. *Experimental Astronomy*, 32:193–316, 2011. doi: 10.1007/s10686-011-9247-0.
- F. Aharonian et al. Discovery of very-high-energy  $\gamma$ -rays from the Galactic Centre ridge. *Nature*, 439:695–698, 2006. doi: 10.1038/nature04467.
- F. A. Aharonian et al. High-energy particle acceleration in the shell of a supernova remnant. *Nature*, 432:75–77, Nov. 2004. doi: 10.1038/nature02960.
- S. Alesker. Description of Continuous Isometry Covariant Valuations on Convex Sets. *Geometriae Dedicata*, 74:241–248, 1999. doi: 10.1023/A:1005035232264.
- S. Alesker. Theory of Valuations on Manifolds: A Survey. *Geometric And Functional Analysis*, 17:1321–1341, 2007. doi: 10.1007/s00039-007-0631-x.
- B. Amedro et al. Current State of Java for HPC. Technical Report RT-0353, INRIA, 2008. URL <http://hal.inria.fr/inria-00312039>.
- G. E. Andrews. *The theory of partitions*. Cambridge University Press, 1998. doi: 10.1017/CBO9780511608650.

- W. B. Atwood et al. The Large Area Telescope on the Fermi Gamma-Ray Space Telescope Mission. *The Astrophysical Journal*, 697(2):1071–1102, 2009. doi: 10.1088/0004-637X/697/2/1071.
- P. Billingsley. *Probability and measure*. Wiley series in probability and mathematical statistics: Probability and mathematical statistics. J. Wiley & Sons, 1995. ISBN 9780471007104.
- B. Boyer. Robust Java benchmarking — IBM developerWorks, 2008. URL <http://www.ibm.com/developerworks/java/library/j-benchmark1/index.html>. [Online; accessed 07-March-2012].
- J. Buckley et al. The Status and future of ground-based TeV gamma-ray astronomy. A White Paper prepared for the Division of Astrophysics of the American Physical Society. *ArXiv e-prints*, 2008. arXiv:0810.0444.
- R. C. G. Chaves. *The extended H.E.S.S. Galactic Plane Survey: Discovering and identifying new sources of VHE gamma-rays*. PhD thesis, Ruperto-Carola University of Heidelberg, 2011. URL <http://www.uni-heidelberg.de/archiv/11756/>.
- M. de Naurois. L’astronomie  $\gamma$  de très haute énergie. Ouverture d’une nouvelle fenêtre astronomique sur l’Univers non thermique. Habilitation thesis, Université Paris VI, 2012. URL <http://tel.archives-ouvertes.fr/tel-00687872>.
- M. de Naurois and L. Rolland. A high performance likelihood reconstruction of  $\gamma$ -rays for imaging atmospheric Cherenkov telescopes. *Astroparticle Physics*, 32(5):231 – 252, 2009. doi: 10.1016/j.astropartphys.2009.09.001.
- G. Dobler et al. The Fermi Haze: A Gamma-ray Counterpart to the Microwave Haze. *The Astrophysical Journal*, 717(2):825, 2010. doi: 10.1088/0004-637X/717/2/825.
- F. Aharonian et al. Observations of the Crab nebula with HESS. *Astronomy and Astrophysics*, 457(3):899–915, 2006. doi: 10.1051/0004-6361:20065351.
- Fermi Science Support Center. Fermi Science Tools Documentation. URL <http://fermi.gsfc.nasa.gov/ssc/data/analysis/documentation/>. [Online; accessed 26-April-2012].

- D. Göring. Analysis of the Poisson Structure of H.E.S.S. Sky Maps with Minkowski Functionals. Diploma thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2008.
- H. Hadwiger. *Vorlesungen über Inhalt, Oberfläche und Isoperimetrie*. Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen mit besonderer Berücksichtigung der Anwendungsgebiete. Springer, 1957. ISBN 3540021515.
- P. J. Huber. *Robust Statistics*. Wiley series in probability and mathematical statistics. Probability and mathematical statistics. Wiley, 1981. ISBN 9780471418054.
- D. Hug, R. Schneider, and R. Schuster. The space of isometry covariant tensor valuations. *St. Petersburg Mathematical Journal*, 19:137–158, 2008. doi: 10.1090/S1061-0022-07-00990-9.
- M. A. Klatt. Morphometric analysis in gamma ray astronomy. Diploma thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2010.
- T.-P. Li and Y.-Q. Ma. Analysis methods for results in gamma-ray astronomy. *The Astrophysical Journal*, 272:317–324, 1983. doi: 10.1086/161295.
- H. Mantz, K. Jacobs, and K. Mecke. Utilizing Minkowski functionals for image analysis: a marching square algorithm. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(12):P12015, 2008. doi: 10.1088/1742-5468/2008/12/P12015.
- J. R. Mattox et al. The Likelihood Analysis of EGRET Data. *The Astrophysical Journal*, 461:396, Apr. 1996. doi: 10.1086/177068.
- P. McMullen. Isometry covariant valuations on convex bodies. In *II International Conference in Stochastic Geometry, Convex Bodies and Empirical Measures : Agrigento, September 9 - 14, 1996*, volume 50 of *Supplemento ai Rendiconti del Circolo matematico di Palermo: Serie II*, pages 259–273. Circolo matematico di Palermo, 1997.
- K. Mecke. *Integralgeometrie in der Statistischen Physik*. Harri Deutsch, 1994. ISBN 3817113803.

- K. Mecke. Integral Geometry in Statistical Physics. *International Journal of Modern Physics B*, 12:861–899, 1998. doi: 10.1142/S0217979298000491.
- K. Mecke. Additivity, Convexity, and Beyond: Applications of Minkowski Functionals in Statistical Physics. In *Statistical Physics and Spatial Statistics*, volume 554 of *Lecture Notes in Physics*, pages 111–184. Springer Berlin / Heidelberg, 2000. doi: 10.1007/3-540-45043-2\_6.
- K. Mecke, T. Buchert, and H. Wagner. Robust morphological measures for large-scale structure in the Universe. *Astronomy and Astrophysics*, 288:697–704, 1994. URL <http://adsabs.harvard.edu/abs/1994A%26A...288..697M>.
- H. Müller. Über Momente ersten und zweiten Grades in der Integralgeometrie. *Rendiconti del Circolo Matematico di Palermo*, 2:119–140, 1953. doi: 10.1007/BF02871682.
- J. Neyman and E. S. Pearson. On the Problem of the Most Efficient Tests of Statistical Hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, 1933. doi: 10.1098/rsta.1933.0009.
- P. L. Nolan et al. Fermi Large Area Telescope Second Source Catalog. *The Astrophysical Journal Supplement Series*, 199(2):31, 2012. doi: 10.1088/0067-0049/199/2/31.
- W. H. Press. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 2007. ISBN 9780521880688.
- G. P. Rowell. A new template background estimate for source searching in TeV  $\gamma$ -ray astronomy. *Astronomy and Astrophysics*, 410(1):389–396, 2003. doi: 10.1051/0004-6361:20031194.
- RRZE. TinyFat cluster — RRZE HPC systems, 2011. URL <http://www.rrze.uni-erlangen.de/dienste/arbeiten-rechnen/hpc/systeme/memoryhog.shtml#tinyfat>. [Online; accessed 07-March-2012].
- L. A. Santaló and M. Kac. *Integral Geometry and Geometric Probability*. Cambridge University Press, second edition, 2004. doi: 10.1017/CBO9780511617331.



- J. Schmalzing and K. M. Górski. Minkowski functionals used in the morphological analysis of cosmic microwave background anisotropy maps. *Monthly Notices of the Royal Astronomical Society*, 297(2):355–365, 1998. doi: 10.1046/j.1365-8711.1998.01467.x.
- G. E. Schröder-Turk et al. Tensorial Minkowski functionals and anisotropy measures for planar patterns. *Journal of Microscopy*, 238(1):57–74, 2010. doi: 10.1111/j.1365-2818.2009.03331.x.
- G. E. Schröder-Turk et al. Minkowski Tensor Shape Analysis of Cellular, Granular and Porous Structures. *Advanced Materials*, 23(22-23):2535–2553, 2011. doi: 10.1002/adma.201100562.
- M. Su, T. R. Slatyer, and D. P. Finkbeiner. Giant Gamma-ray Bubbles from Fermi-LAT: Active Galactic Nucleus Activity or Bipolar Galactic Wind? *The Astrophysical Journal*, 724(2):1044, 2010. doi: 10.1088/0004-637X/724/2/1044.
- The Apache Software Foundation. The Apache Cassandra Project, 2008. URL <http://cassandra.apache.org/>. [Online; accessed 08-March-2012].
- T. C. Weekes et al. Observation of TeV gamma rays from the Crab nebula using the atmospheric Cerenkov imaging technique. *The Astrophysical Journal*, 342:379–395, July 1989. doi: 10.1086/167599.
- Wikipedia. Java performance — Wikipedia, The Free Encyclopedia, 2012. URL [http://en.wikipedia.org/w/index.php?title=Java\\_performance&oldid=478748870](http://en.wikipedia.org/w/index.php?title=Java_performance&oldid=478748870). [Online; accessed 07-March-2012].