A Mirror Alignment Control System for Phase II of the HESS Experiment and A Morphology Study of HESS J1837-069

Dissertation
der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
Stefan Schwarzburg
aus Reutlingen

Tübingen
2012

Tag der mündlichen Qualifikation:	27.07.2012
Dekan:	Prof. Dr. Wolfgang Rosenstiel
1. Berichterstatter:	Prof. Dr. Andrea Santangelo
2. Berichterstatter:	Prof. Dr. Jörn Wilms

Abstract

The work presented in this thesis is separated into two parts: the first describes the development of the mirror alignment control system for H.E.S.S. Phase 2 (MACSII). H.E.S.S. Phase 2 is the extension of the currently four existing telescopes of the H.E.S.S. experiment in Namibia by a fifths, larger telescope (CT5). Building on the principles of the mirror alignment for the existing smaller telescopes, MACSII requires a completely new hardware and software system to match the constraints introduced mainly through the size of and the number of mirrors on CT5.

The second part of the thesis turns from the hardware and software of the mirror alignment system to data analysis. One specific source (HESS J1837-069) has been analysed in regard to the question, whether two separate sources in the observation region can be claimed. The analysis used a program (called LIRA) which has never been used for H.E.S.S. data before. The aim was to assess, if this program proofs potential and can be applied in the area of VHE data especially for blind feature separation. LIRA proofed to be a valid morphology tool. The analysis of HESS J1837-069 revealed that the source region can indeed be separated into individual features. However, the statistical significance of the second hot-spot is not high enough to definitely claim a detection.

Zusammenfassung

Die vorliegende Dissertation umfasst zwei Teile: Teil 1 beschreibt die Entwicklung eines Spiegelkontrollsystems für H.E.S.S. Phase II. H.E.S.S. Phase II. ist die Erweiterung eines bereits bestehenden Teleskopexperiments im Hochland von Namibia. Das Experiment wird momentan um ein fünftes, deutlich größeres Teleskop (CT5) ergänzt, das sich in der Aufbauphase befindet und neue Herausforderungen mit sich bringt. Unter anderem wird ein geeignetes Hardware- und Softwaresystem benötigt, um die große Anzahl an Spiegeln von CT5 auszurichten.

Teil 2 der Dissertation wendet sich der Datenanalyse zu. Eine bestimmte H.E.S.S. Quelle (HESS J1837-069) wurde mit Blick auf die Frage untersucht, ob zwei voneinander getrennte Quellen in der Beobachtungsregion erkennbar sind. Die Analyse verwendete hierzu ein Programm namens LIRA, welches noch nie zuvor auf H.E.S.S. Daten angewendet wurde. Ziel war es herauszufinden, ob dieses Programm im Bereich der Datenanalyse der Hochenergie-Astrophysik einsetzbar ist – speziell für das voraussetzungsfreie Separieren von Quellregionen. LIRA stellte sich als ein valides Analyseprogramm heraus. Die Analyse von HESS J1837-069 zeigte, dass in der Quellregion tatsächlich zwei getrennte Quellen sichtbar sind. Allerdings erwies sich die statistische Signifikanz der zweiten Quelle als nicht hoch genug, um definitiv eine Detektion zu behaupten.

Contents

Chapter 1

General Introduction

The field of VHE Astronomy is a rather new and quickly developing field in astronomy. Up to now there are only few telescopes operating in this energy range. The H.E.S.S. collaboration (see Section 3.2) is one of a few international collaborations addressing the observation of cosmic γ-rays in the 100 GeV to 100 TeV energy range.

When this thesis project was started in 2005, H.E.S.S. already worked with four telescopes in Namibia (CT1-4) and planned to extend the array to five telescopes (CT5). The aim of the thesis thus was on the one hand to provide the H.E.S.S. collaboration with a mirror alignment system for the fifths telescope, and on the other hand to work with data obtained with from the already existing and operating telescopes. Accordingly, the thesis consists of two parts:

The first part is hardware as well as software orientated and introduces the mirror alignment control system developed as part of this thesis. It gives an introduction to the H.E.S.S. project, to the observation techniques it uses and then describes and discusses the ideas of the alignment system. The following topics are of central interest to this part: the hardware development including FPGAs, the Suzaku board used, the hardware implementation and testing, as well as the software development on a uClinux kernel, the communication between the components, the central control software, and the alignment algorithm. The first part concludes with an outlook on the future project CTA, which will follow H.E.S.S. and Magic and which will probably use a comparable alignment system.

The main reason why CT5 needed a new alignment system is, that the fifths telescope is much larger than the operating telescopes CT1-4. The size asks for many new requirements in the alignment system, ranging from a different focus on lightning protection to the ability to move several mirrors at the same time in order to speed up the alignment process. These requirements resulted in a decentralized motor control system with embedded computers in the telescope structure. Programming these embedded computers or SoCs (System-on-a-Chip) and extending the FPGAs which build the basis of these embedded computers, proved to be the most time consuming task of this thesis.

At the moment CT5 is built. The steel structure is finished, the mirrors will be installed during the summer of 2012 and the mirror alignment will take place after that.[1]

The second part of the thesis turns to data analysis with already existing H.E.S.S. data. One specific source (HESS J1837-069) has been analysed in regard to the question, whether

[1] Based on the current estimate, the first alignment will take place in July 2012 and a second alignment campaign will be held during September 2012.

two separate sources in the observation region can be claimed. This is in itself an interesting question. However, additionally the analysis used a program (called LIRA) which has never been used for H.E.S.S. data before. The aim was to assess, if this program proofs potential and can be applied in the area of VHE data especially for blind feature separation.

The second part starts with an introduction to VHE γ-ray physics and then turns to the characteristics of LIRA. Especially the correct mathematical treatment of low-count data and the deconvolution properties were a reason to use LIRA as a tool to infer source separation and feature significance in low-count observations. Chapter 10 goes into details of testing LIRAs capabilities with simulated sources, whereas Chapter 11 finally applies LIRA to the mentioned H.E.S.S. source.

The main task for this second part was to find an unbiased method which allows to quantify whether the source region splits into several sub-features and to find a measure for the detection significance for these features. It it not possible to answer a question like this with the standard H.E.S.S. approaches. These have to resort to shape assumptions for fitting techniques (most prominently 2D-gaussian distributions and χ^2- or maximum likelihood fits) or they lose spatial information by the use of re-binning and slicing techniques with additional fits (again 1D-gaussians and χ^2-fits).

The chosen approach involving LIRA and a simple thresholding technique for feature separation seems promising, since LIRA incorporates many useful features: it takes the PSF and an exposure into account while it forward-fits a background and a multi-scale representation to the source region with a Markov-chain Monte Carlo method. The full mathematical treatment of low-count pixel values in the Bayesian framework of LIRA results not only in a best-fit for the source, but in a full distribution of possible source intensities which can be used to quantify knowledge or certainty about the source shape. To assess feature significance, the best-fit reconstructed model of a LIRA analysis is separated into clearly distinct features. These features are then removed from the model for individual reference simulations, in order to gain simulated observations from the individual null-models. Null-models follow the hypotheses that the features are just fluctuations of the background and the surrounding features.

The main result of the second part of the thesis is, that LIRA, although it comes with some disadvantages, can be a valid alternative to other analysis tools in VHE morphology analysis. The analysis of HESS J1837-069 itself reveals that the source region can indeed be separated into individual features, however, the statistical significance of the second hot-spot is not high enough to definitely claim a detection.

Chapter 2

An Introduction to Imaging Atmospheric Cherenkov Telescopes

Very high energy (VHE) γ-ray astronomy is a relatively new field dealing with the observation of photons with energies above $\sim\ 100\,\mathrm{GeV}$. Photons with high energies (>30 MeV) and very high energies do not reach the surface of the Earth, but interact with molecules in the atmosphere. *Direct* measurements of these γ-rays can therefore only be performed at large heights (>14 km). Satellites like INTEGRAL and Fermi do so and are very successful examples of this strategy. At higher energies, the γ-ray flux declines steeply. Therefore, to detect such photons, larger collection areas or interaction volumes are needed. The Fermi satellite for example is sensitive in the energy range from 20 MeV up to 300 GeV (see Baldini and the Fermi LAT Collaboration 2011). The LAT detector on Fermi already has a volume of 1733 x 1733 x 970 mm (see Moiseev 2008). To observe photons with even higher energies other methods besides satellites must be used. This is where the *indirect*, ground based methods of VHE γ-ray observation begin.

When cosmic rays enter the atmosphere, they do not simply vanish. The interactions with molecules that keep the particles from reaching the ground result in many secondary particles (described in Chapter 2.1). Some of these particles, like the optical photons, reach the ground, and can be observed with optics sensitive in the optical and UV energy range. This technique is *indirect* insofar as it relies on secondary particles in order to obtain insight into the primary particles and it is called ground based atmospheric Cherenkov technique. This method is described in Chapter 2.2.

The experiment of VHE astronomy which will be the focus of this thesis is the H.E.S.S. experiment. This experiment is already running with an array of four telescopes in Namibia, operated by an international collaboration (see Chapters 3.1 and 3.2). More information on H.E.S.S. and specifically on the mirror alignment control system of H.E.S.S. Phase 2, which was developed within this PhD project, follow the introduction to the basics of VHE γ-ray observations.

2.1 The Characteristics of Air Showers

There are two main categories of air showers which can be observed with Cherenkov telescopes.

The first ones are hadronic air showers, which are created by the hadronic component of cosmic rays, like protons or stripped atomic nuclei. These are considered background in VHE γ-ray experiments, because charged particles do not point to the region of their origin, but are taken off the straight line by intergalactic, interstellar or interplanetary

magnetic fields.
The second category of air showers observable with Cherenkov telescopes are electron-photon cascades or electro-magnetic air showers, which stem from photons or electrons. The ones created by photons are considered the signal in experiments like H.E.S.S..

2.1.1 Eletro-magnetic Showers

VHE γ-rays can convert into e^-e^+-pairs in the Coulomb field of atomic nuclei in the atmosphere. These electrons (the same argument applies to positrons as well) can then be deflected by nuclei. During these interactions they emit Bremsstrahlung photons. The photons originating from the Bremsstrahlung can then in turn create other e^-e^+-pairs. The exponential growth of the number of particles in the air shower continues until the mean energy drops below the critical energy $E_c \sim 80\,\mathrm{MeV}$. At that energy, the electrons lose more energy by ionisation than by Bremsstrahlung, and the number of particles in the shower decreases.
The same cascade that stems from the electrons which are created in the first interaction between γ-rays and nuclei can of course also be initiated by high energy electrons in the cosmic rays.
To estimate the length of an electro-magnetic shower, the mean free path of electrons in the atmosphere is important. The mean free path is the distance after which all but $1/e$ of the initial energy E_0 is lost. In the case of Bremsstrahlung, this is called the radiation length X_0, typically measured in g $*$ cm^{-2}. The energy loss of an electron due to Bremsstrahlung after travelling a distance x in matter is:

$$-\frac{dE}{dx} = \frac{E_0}{X_0} \tag{2.1}$$

In air, a typical value for the radiation length is $X_0 = 37.2\,\mathrm{g} * \mathrm{cm}^{-2}$.
The lateral development of the shower is mainly influenced by Coulomb scattering, in which photons and electrons are multiple times scattered off nuclei under small angles. Bremsstrahlung and pair production also contribute to the lateral spread of the shower particles.
If high energetic charged particles traverse through matter, as the electrons in the air shower do, another kind of process is important: the Cherenkov radiation. This radiation was discovered by Pavel Alekseyevich Cherenkov, a Soviet physicist who won the Nobel Prize in physics in 1958 for his discovery. The Cherenkov radiation also gives the name to the main technique (Imaging Atmospheric Cherenkov Technique) used in ground based TeV astronomy today.
When the charged particles with a velocity $v = \beta c$ higher than the phase velocity of light $\frac{c}{n}$ in the ambient matter pass through dielectric matter with refraction index n, molecules become excited and emit light. A part of this light is emitted in a coherent way and forms a wavefront in a fixed angle Φ relative to the path of the particle:

$$\cos\Phi = \frac{1}{\beta n} \tag{2.2}$$

Combining this angle with the lateral shower expansion leads to an opening angle of the Cherenkov light of up to 1° with respect to the initial γ-ray path. With a typical height of the first interaction of 8 − 14 km and an observation height of 2000 m asl[1], the Cherenkov light cone resulting from a vertically penetrating γ-ray has a radius of about 120 m on the ground.

The number of photons emitted per path length x between λ_1 and λ_2 is

$$\frac{dN}{dx} = 2\pi\alpha z^2 \int_{\lambda_1}^{\lambda_2} (1 - (\beta n(\lambda))^{-2})\lambda^{-2} d\lambda \qquad (2.3)$$

with α the fine structure constant and z the charge number (Bernlohr 2000). This results in a maximum intensity of the Cherenkov light initially in the UV, but due to absorption, the maximum of the observable light on the ground is at a wavelength of 300-400 nm (Kohnle et al. 2000).

Other basic properties of the shower can easily be derived from a simplified model by Heitler (1954) as can be seen in Figure 2.1. In this model, only the dominant processes, pair production and Bremsstrahlung, are considered. Other simplifications include setting the radiation length of both processes to the same value X_0 and the assumption that the energy is divided into equal parts when particles are created.

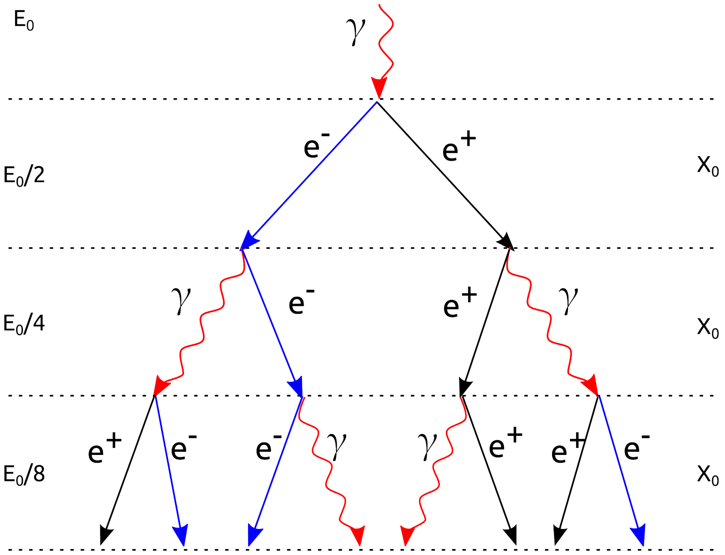


Figure 2.1: Simplified model of the development of an electromagnetic shower according to Heitler (1954). A photon with the initial energy E_0 decays via pair production into an electron and a positron after an interaction length of X_0. The products emit a photon through Bremsstrahlung. This continues until the mean energy of the particles drops below a critical energy E_c∼ 80 MeV.

[1] asl: above see level; The H.E.S.S. experiment is located at 1800 m asl (see Chapter 3).

This leads to the following formula for the number of particles:

$$N(x) = 2^{x/X_0} \tag{2.4}$$

where x is the distance travelled along the shower axis, measured in g/cm^{-2}. Given the energy E₀ of the primary photon, the energy of a particle is then

$$E(x) = \frac{E_0}{N_x} = E_0 \times 2^{-x/X_0} \tag{2.5}$$

This formula can be combined with the critical energy E_c to derive the maximum number of particles in the shower:

$$E_c = E_0 \times 2^{-x_{max}/X_0} \Rightarrow x_{max} = \frac{\ln{(E_0/E_c)}}{\ln 2} \times X_0 \tag{2.6}$$

which leads to the number of particles at the shower maximum of

$$N_{max} = 2^{x_{max}/X_0} = \frac{E_0}{E_c}. \tag{2.7}$$

The two main predictions of this toy model are therefore: the number of particles in the shower is proportional to the initial energy of the γ-ray and the vertical size of the shower grows exponentially with the initial energy. These characteristics do not change much, when better models are used which take for example absorption effects into account (For detailed research see e.g. Montanus 2012, Rossi and Greisen 1941). For a primary γ-ray of 1 TeV energy, about 100 Cherenkov photons per square meter reach a height of 2000 m asl.

2.1.2 Hadronic Showers

If hadronic cosmic rays, mainly protons and stripped atomic nuclei, hit the atmosphere, they are inelastically scattered. This interaction produces several kinds of particles like mesons (kaons, pions), nucleons (neutrons, protons) and hyperons (Δ, Λ, Σ, Φ) which start a hadronic cascade.

Important for the Cherenkov technique are the neutral pions (π^0) which almost instantly decay into γ-pairs, thereby starting electro-magentic subshowers as described above in the case of initial γ-rays.

To distinguish between the γ-induced and the hadronic induced showers, the difference in the morphology of the particle shower and therefore of the Cherenkov light distribution on the ground has to be used.

The difference between the shower types can be seen in Figure 2.2 and 2.3 by example of a simulated 1 TeV proton and a simulated 300 GeV photon. The diffuse shapes of the Cherenkov light of the hadron induced showers can be used to remove the hadronic background from the recorded data. Monte Carlo Simulations like the ones shown in

Figures 2.2 to 2.5 (Bernlöhr 2000) using the CORSIKA package[1], can be the basis for such discrimination.

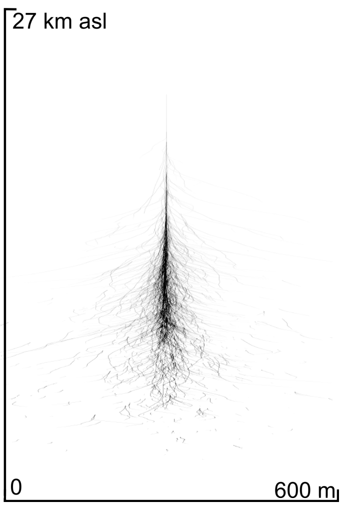


Figure 2.2: Simulation of the air shower development originating from a 300 GeV photon (taken from Bernlöhr 2000).

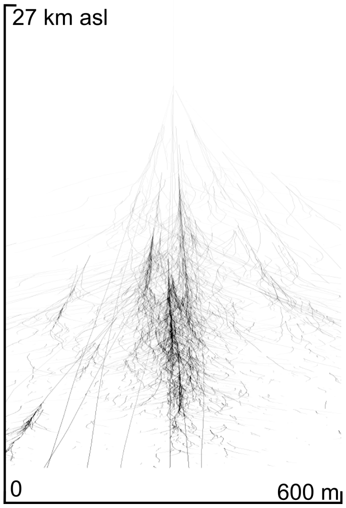


Figure 2.3: Simulation of the air shower development originating from a 1 TeV proton (taken from Bernlöhr 2000).

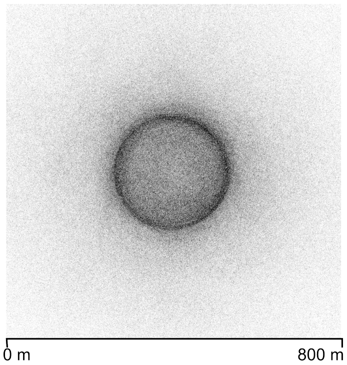


Figure 2.4: Simulated light distribution on the ground originating from a 300 GeV photon (taken from Bernlöhr 2000).

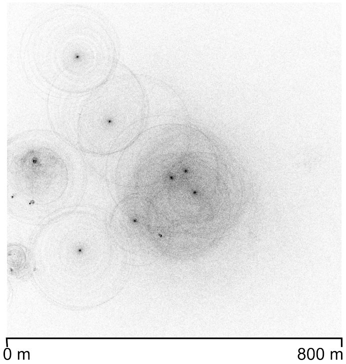


Figure 2.5: Simulated light distribution on the ground originating from 1 TeV proton (taken from Bernlöhr 2000).

2.2 *The Stereoscopic Cherenkov Technique*

Imaging Atmospheric Cherenkov Telescopes (IACTs) are instruments which observe the Cherenkov light of air showers induced by incident γ-rays. They focus this secondary light with mirrors on a dish or reflector[2] onto a detector in the focal plane.

[1] For more information about CORSIKA, see `http://www-ik.fzk.de/corsika/technicalities.html` and the original paper (Heck et al. 1998).

[2] The terms dish and reflector are used synonymously throughout this thesis, because every time the dish is referenced, the steel structure and the mirrors attached to it are meant.

The detectors of current atmospheric Cherenkov telescopes like H.E.S.S. (Kohnle et al. 2000), MAGIC (Borla Tridon et al. 2009) or Veritas (Weekes et al. 2002) are cameras composed of photomultiplier tubes (PMTs). These can be read out fast enough to resolve the individual showers.

In TeV astronomy three parameters related to a photon are generally of interest: The arrival time, the energy of the photon and its direction. While the arrival time is relatively easy to measure, the other two parameters have to be reconstructed from the actual observation of the Cherenkov light cone due to the indirect measurement technique of the telescopes.

2.2.1 The Hillas Parameters

The analysis of H.E.S.S. observations starts with the shower images. These are captured by PMT cameras on the telescope. Each PMT camera delivers an image of the light cone, which is roughly an elliptically shaped spot surrounded by night sky background (NSB). At first, the spot is separated from the NSB by a combined thresholding and clustering technique (see Aharonian et al. 2006a). Each spot image is then reduced to a number of parameters defined by Hillas (1985). These so called Hillas parameters are used to describe the ellipse. A sketch of a cleaned PMT image with the parameters can be seen in Figure 2.6. The parameters are: the length L, the width W, the size (which means the total image amplitude), the nominal distance D (which is the distance between the centre of the camera and the image centre of gravity), the azimuthal angle of the main axis of the ellipse φ, and the orientation angle α.

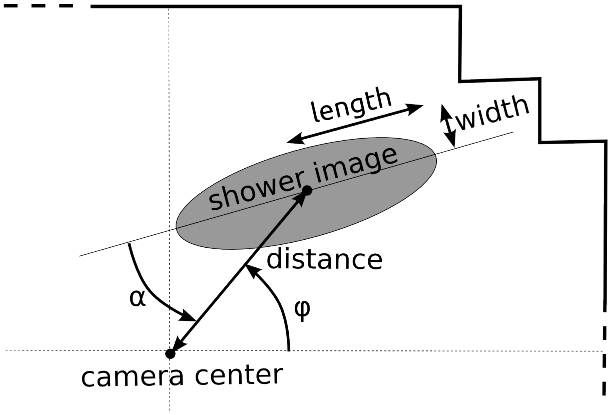


Figure 2.6: Definition of the Hillas parameters by the example of an idealized PMT camera image.

2.2.2 The Direction Reconstruction

It is possible to derive the direction of the primary particle from the Hillas parameters. In a single telescope configuration, the direction can be estimated from the size and the length of the ellipse using either look up tables or ad-hoc analytical functions (see de

Naurois 2006). But with the simple symmetric parametrization, there is an ambiguity in the direction of the primary particle which leads to two possible directions, one to each side of the center of gravity of the shower image. To prevent this ambiguity, third order moments would have to be introduced as additional parameters.

Another solution to avoid degenerate solutions for the direction reconstruction is to use more than one telescope to stereoscopically observe the showers. The solution for the shower impact and the direction of the primary photon can then be calculated from the intersection of the main axis of the ellipse in the different shower images. For more than two telescopes, a weighted mean can be used. This technique is illustrated in Figure 2.7. This stereoscopic Cherenkov technique was introduced with HEGRA (Daum et al. 1997) and is today in use at all new Cherenkov telescopes (e.g. H.E.S.S., MAGIC, Veritas).

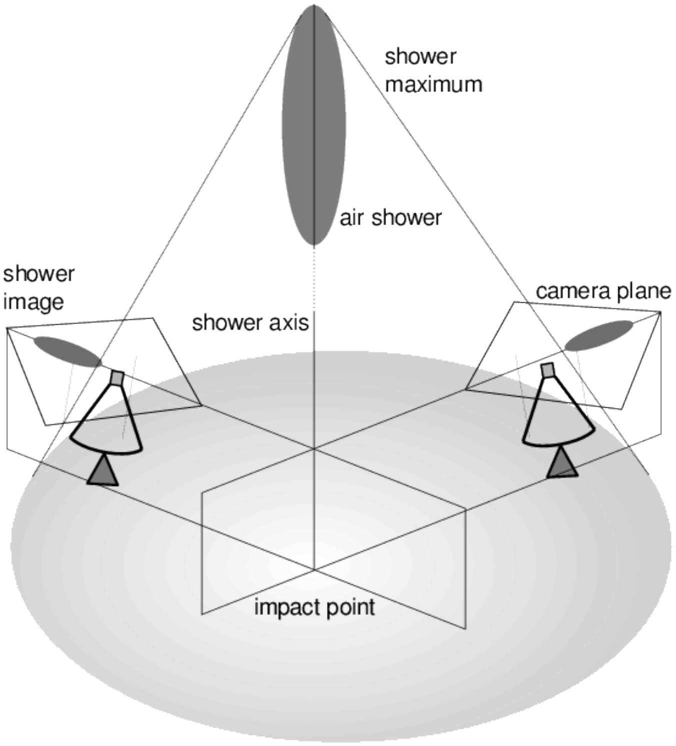


Figure 2.7: Reconstruction of the shower direction using the intersection of the main axis of the shower image from two telescopes.

2.2.3 *The Energy Reconstruction*

As described in Chapter 2.1.1, the maximum number of particles in the electro-magnetic air shower is proportional to the primary γ-ray energy. Therefore the number of Cherenkov photons can be used to estimate the energy of the primary particle, under the assumption that the initial particle was a photon. Monte-Carlo simulations of extended air showers are used to create more realistic particle distributions as functions of the primary energy and its direction.

It is often said that imaging atmospheric Cherenkov telescopes use the whole atmosphere as a detector. From this point of view, the effective detection area of a telescope is about $50000\,\mathrm{m}^2$.[1] This however redefines the meaning of the word 'detector', which originally means 'an object or machine which has been invented to find particular substances or things, or measure their level'.[2] This is clearly not the case with the atmosphere – a fact that can be seen in the origin of the biggest systematic uncertainty of the energy reconstruction: the density, temperature, and humidity profiles of the atmosphere, which are not known a priori, change over time and therefore have to be monitored constantly along with the showers themselves. While the density profile affects the height of the shower maximum, clouds and particles in the air affect the absorption of the Cherenkov light.

Other systematic uncertainties arise from the optical system and the camera. To calibrate the H.E.S.S. camera, the so called single photo - electron response is measured. The response is mainly affected by two components: the individual photomultiplier tubes and the light collectors in front of the PMTs (Winston cones). The calibration is done with a LED system in front of the PMT and in the dish (Aharonian et al. 2004).

The optical system, that means the individual mirror reflectivity, the mirror alignment, and the shadowing from the telescope structure can be calibrated with myon runs. These are observations which analyse selectively the Cherenkov light of myons passing close to the camera. The Cherenkov light of these particles forms very characteristic rings (myon rings) which have the special property that the number of Cherenkov photons can be calculated based on few geometric parameters of the myon (see Bolz (2004) for a detailed description). If the number of Cherenkov photons is known and the camera response is understood, then the optical efficiency of the telescope can be determined in this way.

Based on all this input and with the help of Monte-Carlo simulations, look-up tables are created that map the Hillas parameters to the true energy of the primary photon. If the energy reconstruction differs greatly between the telescopes (> 25%), the event is rejected. Other selection criteria are for example unstable weather conditions, which are monitored through the trigger rates of the telescopes, or a distance of the event of more than 2° from the center of the FoV (see Aharonian et al. 2006a).

2.2.4 The Background Rejection

The simulated images of air showers in Figure 2.2 and 2.3 show that hadronic and electromagnetic shower images differ. To quantify this difference and use it as a discrimination

[1] See: `http://www.mpi-hd.mpg.de/hfm/HESS/pages/about/telescopes/#cherenkov`

[2] According to the Cambridge Dictionary online: `http://dictionary.cambridge.org/dictionary/british/detector?q=detector` and `http://dictionary.cambridge.org/dictionary/british/device_1?q=device`

factor for background suppression, so called *scaled parameters* are calculated from the shower images. The parameter *mean reduced scaled width* is defined as

$$MRSW = (\sum_{tel}(w - \langle w \rangle)/\sigma_w)/N_{tel} \tag{2.8}$$

It describes the comparison of the observed parameter w (shower width) with the expected width $\langle w \rangle$, which is taken from simulations of the mean width, in units of the standard deviation σ_{w}. This value is averaged over all telescopes that observed the shower. The *mean reduced scaled length* (MRSL) is defined accordingly (see Aharonian et al. 2006a). When comparing the distribution of MRSW and MRSL of simulated protons with the distribution of simulated γ-rays, thresholds can be defined that provide different probabilities of selecting only γ induced showers and rejecting hadronic showers. The higher this probability has to be, the more events have to be rejected. Depending on the analysis goal, different so called *selection cuts* are in use. Apart from selecting based on the mean scaled parameters, the image amplitude as well as distance to the center of the FoV are also taken into account.

The *hard cuts* that were used in the morphological analysis in Part II, are tuned to sources with a flux of 1% of the Crab and a Γ of 2 (see Aharonian et al. 2006a). These cuts additionally provide a narrower PSF than *standard cuts* as the thresholds on the intensity of the shower images (200 photo electrons compared to 80 in standard cuts) select better reconstructed events. Selecting on spatial reconstruction quality is an obvious choice for morphological analysis.

Alternative Methods

A different reconstruction technique which does not reduce the shower image to a small set of parameters but uses the full camera information is in use by some groups in the H.E.S.S. collaboration (de Naurois and the H. E. S. S. Collaboration 2003). The approach is called 'Model Analysis' and is based on comparing each pixel with a set of templates, that were created from a shower development model. A density function is created which describes the probability of finding a signal in a pixel. From this, a log-likelihood function is maximised to estimate the energy and direction of the primary particle. This technique works in the same way for a single telescope and for an array of stereoscopic observing telescopes.

Yet another option could be to use time resolved shower images. Studies with the Hegra telescopes have shown, that the shower image has a timing gradient along the major axis (HEGRA Collaboration et al. 1999) which varies with the core distance. From this information it would be possible to derive the shower direction. This technique is described in Stamatescu et al. (2008) for a possible future IACT called *TenTen* which focuses on a higher energy range (10-100 TeV) where this technique could help improving the shower reconstruction – at least for larger distances of the shower core.

Chapter 3

The H.E.S.S. Experiment

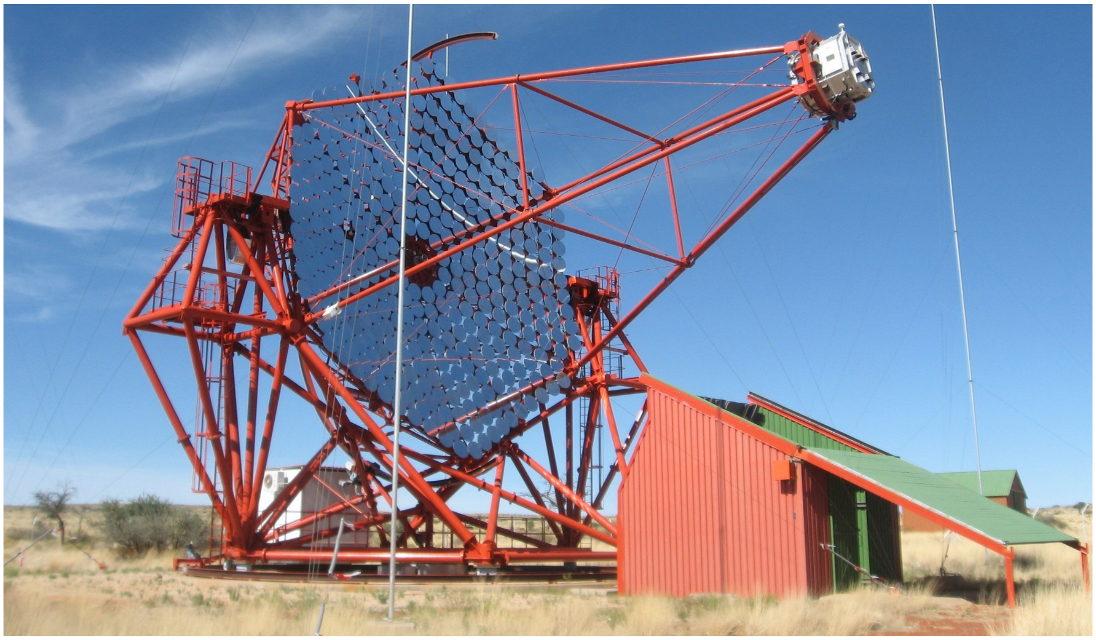


Figure 3.1: One of the four currently operating H.E.S.S. telescopes. The telescope is 'parked out', which means that the photomultiplier camera is out of its safety position in the hut. This is normally only done during nights, before observations start.

In 2004, about 10 years after the first planning steps, the four H.E.S.S. telescopes started regular operation as a stereoscopic Cherenkov Telescope Array.

The name H.E.S.S. has two meanings: on the one hand it is an acronym for **H**igh **E**nergy **S**tereoscopic **S**ystem, which describes the operation mode and observation technique of the system. On the other hand it is a reference to Victor Franz Hess, an Austrian-American physicist who won the Nobel prize in physics in 1936 for the discovery of cosmic rays.

Figure 3.2: V. Hess in front of the Hafelekar-Labor after receiving the Nobelprize (Nov. 1936) and after the balloon flight which lead to the discovery of cosmic rays.[1]

3.1 The Location of the H.E.S.S. Experiment

The H.E.S.S. telescopes are located in the Khomas highlands of Namibia (23°16'18" S, 16°30'00" E) at a height of 1800 m asl. The H.E.S.S. site is about 100 km from Windhoek, the capital of Namibia (see Figure 3.3). The place was chosen for several reasons:
Like for most telescopes, high altitudes reduce the atmospheric absorption (the first interaction height of γ-rays with the atmosphere is much higher, at about 14 km). Additionally, in Namibia the night sky background component due to artificial light is strongly reduced, mainly because the Republic of Namibia has a very low population density, of only about 2.6 people per km^2.

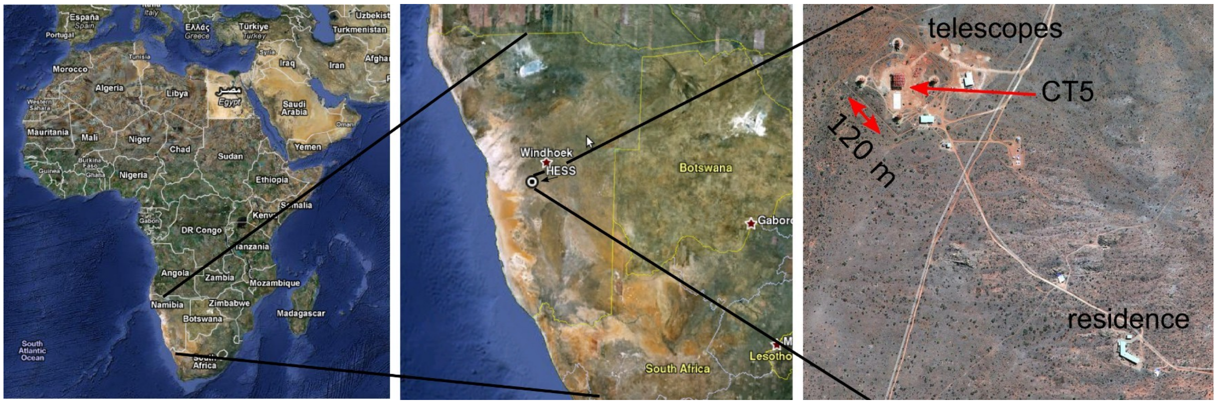


Figure 3.3: Location of the H.E.S.S. telescopes. (Maps: maps.google.com)

The third advantage besides height and reduced background light is the good weather

[1] Photos: `http://physik.uibk.ac.at/hephy/Hess/Steinmaurer.html`

conditions on the site, with a relative high number of about 1000 cloudless, dark hours[1] per year (Wiedner 1998).

Apart from these reasons the location on the southern hemisphere has the advantage of a good visibility of the Galactic center. The observation of the Galactic center region (Aharonian et al. 2006) has been a key goal of the H.E.S.S. collaboration since the beginning and scans of the Galactic center and the Galactic plane have been conducted very successfully (see e.g. Chaves et al. 2008).

3.2 The H.E.S.S. Collaboration

Scientists from all around the world are contributing to the H.E.S.S. experiment. The current member list includes institutes from Germany, France, the United Kingdom, Namibia, South Africa, the Czech Republic, Ireland, Armenia, Poland, Australia, Austria, Sweden and some other associated members.

Major tasks are divided between different working groups. They are concentrating for example on the combination of H.E.S.S. results with results from other telescopes like XMM-Newton (Multiwavelength Observations), on Pulsars, Pulsar Winds, Plerions, and SNR, or on Astroparticle Physics and Exotic Phenomena. The collaboration meets in these working groups as well as twice a year in full member conferences.

The telescopes have been planned and built by the collaboration. They are operated and maintained by staff working permanently on the site as well as by experts from the member institutions. The observations are also conducted by members of the collaboration, which take turns in monthly shifts on the site.

Since the H.E.S.S. experiment is not an observatory but fully financed by its members, the data analysis rights stay within the collaboration.

3.3 The H.E.S.S. Phase 1 Telescopes

The four currently operating telescopes of the H.E.S.S. project are positioned in the corners of a square of 120 m length. This size was chosen as a compromise between a better event reconstruction and the aim to capture as many air showers as possible. The event reconstruction becomes more precise, if the telescopes imaging a shower are further apart from each other. As shown in Chapter 2.1.1, the distance of 120 m was chosen, because the typical size of the Cherenkov light cone at 1800 m asl is about this size in radius.

The main structure of the H.E.S.S. Phase 1 telescopes is made of steel, to prevent pointing dependent deformations. The mounting is an altitude-azimuth (so called 'alt-az') system (see Bernlöhr et al. 2003).

The slewing speed of the telescope is about $100^\circ/\text{min}$ in both elevation and in azimuth (Bernlöhr et al. 2003) where the whole telescope rotates on a rail drive system as can be seen in Figure 3.4.

[1] Photometric dark hour means that no clouds above an altitude of 30° are visible. Additionally the Sun has to be $\leq 18^\circ$ below the horizon while the moon is below the horizon as well.

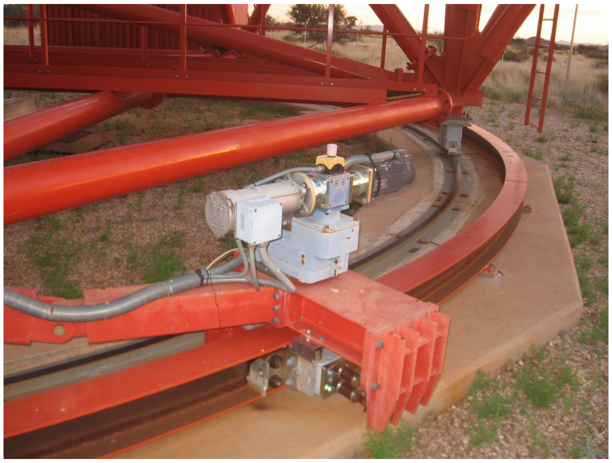


Figure 3.4: Azimuthal drive system of a H.E.S.S. telescope.

The reflector follows a Davis-Cotton design with an overall mirror size of $107\,\mathrm{m}^2$ and has a diameter of $d \sim 15\,\mathrm{m}$ (see Bernlöhr et al. 2003).
The photomultiplier camera is positioned at 15 m from the reflector which leads to an $f/d \sim 1.2$. The H.E.S.S. Phase 1 telescopes have a FoV of 5°to be able to scan large parts of the sky, but also to observe extended emission regions. The photomultiplier camera (see Figure 3.5) consists of 960 individual photomultiplier tubes (PMTs) where each PMT acts as a pixel that corresponds to $0.16°$ of the sky.

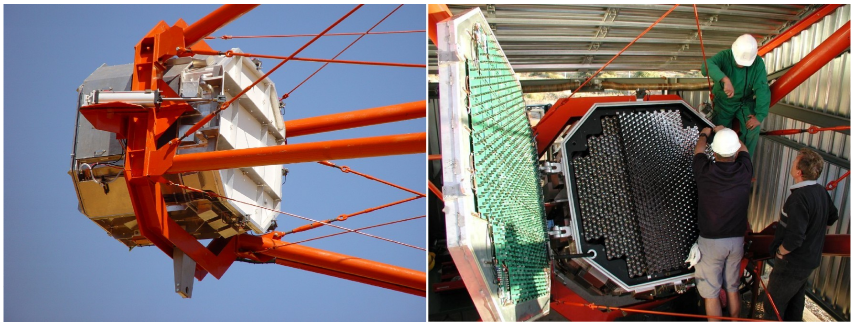


Figure 3.5: H.E.S.S. photomultiplier camera.

In front of each pixel is a Winston cone, which is used to turn the circular shapes PMT openings into a hexagonal shaped light collecting area which can cover the complete surface at the camera plane. The Winston cones are also used as collimators which keep straylight or light coming from larger angles off the PMTs. Sixteen of these pixels form a so called ‘drawer’ which has its own high voltage supply and all electronics needed for the data conversion and readout.

3.4 Phase 2 of the H.E.S.S. Experiment

For the last few years, the H.E.S.S. collaboration has been planning and building the extension of the currently four telescopes (CT1 - CT4) by a fifth much bigger telescope (CT5). All five telescopes together are referred to as H.E.S.S. Phase 2.
CT5 will be placed in the middle of the square that is formed by the other four telescopes. It will be the largest Cherenkov telescope built so far with a reflector size of $600\,\mathrm{m}^2$.
The two goals of H.E.S.S. Phase 2 are an increased sensitivity of the system in the energy regime of the current array (100 GeV - 100 TeV) as well as a lower energy threshold well below 100 GeV.
Originally, Phase 2 was planned as an array of twelve additional telescopes of the same type as CT1-CT4. But building a single larger telescope offered the opportunity to observe in an energy range which overlaps with the Fermi satellite, operating from 20 MeV to more than 300 GeV (Ritz 2007).

3.4.1 The Telescope Structure of CT5

The telescope structure is again built as a steel space-frame structure which can rotate in azimuth on a circular rail system of 36 m diameter (Deil et al. 2008). Two elevation bearings connect the base structure with the dish and the PMT camera quadrupod (see Figure 3.6). The telescope will have an azimuthal range of 540° with a drive speed of 200°/min. The possible elevation ranges from -32°, which corresponds to the park position, to 180°which is normally only used during maintenance operations. The normal elevation during observations stops at 90°. The elevation speed is 100°/min. Both drive systems will each be operated by four 24 kW servo motors (Deil et al. 2008). The positioning accuracy of the drive system is 10”, measured directly at the telescope axes.

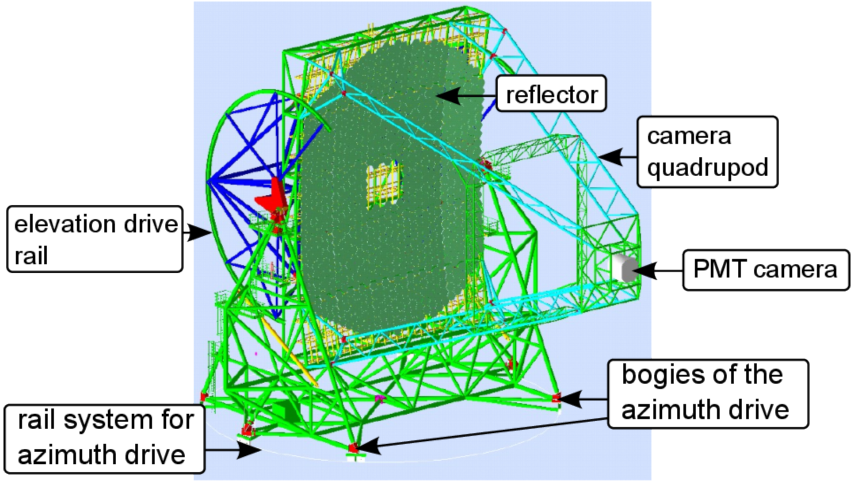


Figure 3.6: H.E.S.S. Phase 2 structure

3.4.2 The Camera

The PMT camera of H.E.S.S. Phase 2 follows the same design of the H.E.S.S. I camera. The size of the PMTs remains the same, but due to the increased focal length of 35 m, the angular diameter of a pixel is reduced to $0.07°$ (Horns and the HESS Collaboration 2007). The camera has a diameter of ~ 2.5 m and contains 2048 pixels packed in 128 drawers. These drawers are all identical and can be removed individually. All electronic components for readout, trigger and networking as well as the power supplies for the PMTs are stored on the back of the camera. The heat produced by that system (about 10 kW) is dissipated by a vortex-cooling system inside the camera (see Vincent et al. 2005).
On the front and on the backside of the camera are pneumatic operated lids. The frontside lid, which can be seen in Figure 3.7 will be used by the mirror alignment system of CT5 as a diffusing screen (see Chapter 4).

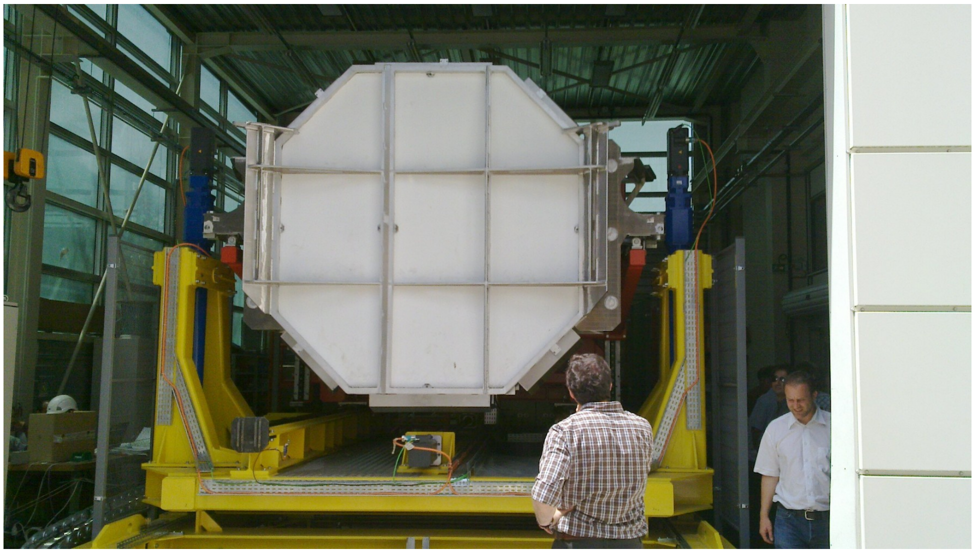


Figure 3.7: The PMT camera for H.E.S.S. Phase 2. Mainly visible is the frontside lid, which will be used by the mirror alignment system to take pictures of the spots cast by the individual mirrors facets. The photo was taken in Annecy (France), where the automatic camera loading and unloading was tested.

Due to the size of the camera and the reduced lightning protection of CT5, special measures were taken to protect the camera. Unlike in H.E.S.S. Phase 1, the shelter of the camera is now a completely separate building into which the camera is transported via rails at the end of each night or if the weather conditions are likely to worsen during an observation. While the camera is unloaded and safely stored in the storage building, a dummy camera with the same weight is installed in its place. This is done to avoid stress on the telescope structure when the ~ 3 t of the camera are missing.
To assure that the camera is at the correct position when it is installed at the telescope at the beginning of each night, an autofocus system was developed, which can change the distance of the camera to the dish.

3.4.3 The Reflector

The reflector of CT5 has a parabolic shape with a focal length of 36 m. The mirror support unit is a steel construction of rectangular beams of 32 m x 24 m size. It is divided into 25 identical panels welded onto the underlying structure. Each panel supports up to 42 mirrors. At some positions at the edges and at the center, mirrors are left out on purpose, either to increase the optical performance (Cornils et al. 2005), or to leave space for additional technical equipment like the Sky- and the Lid-CCD.

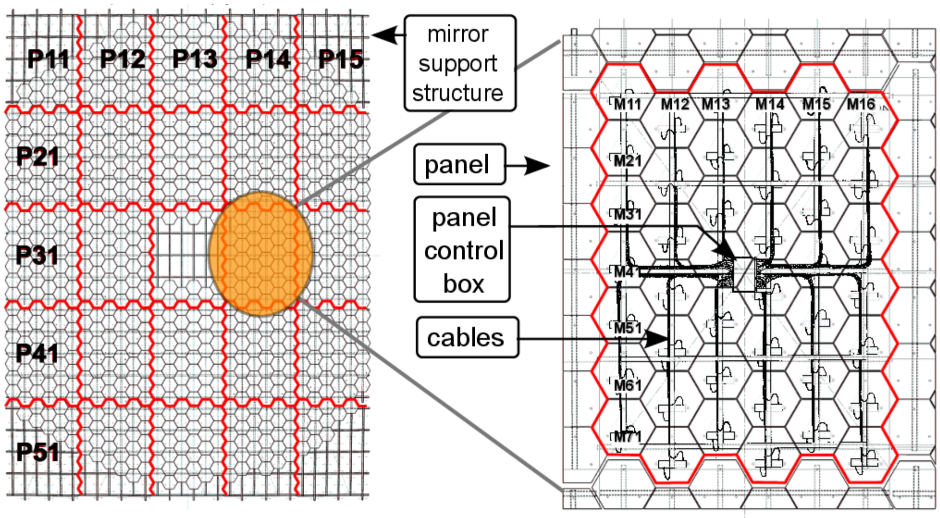


Figure 3.8: The mirror support unit is divided into 25 identical panels. Each panel supports up to 42 mirrors. The panel and mirror numbering scheme of MACS II is shown. A sketch of the cabling and the position of the panel control box is overlayed.

The reflector of the H.E.S.S. Phase 2 telescope will consist of up to 980 mirror tiles[1], each with a flat-to-flat size of 90 cm. These mirror tiles are connected with the mirror support structure on three points, two of them are motor driven actuators. To power these motors, cables are run from the panel control boxes to each motor. Due to the number of mirrors, this requires $\sim$ 7 km of cables. The cables and the actuators were installed on the telescope in November and December 2011 by a team from the IAAT. The actual alignment of the CT5 mirrors is planned for mid 2012.

With the help of the motor driven actuators, the direction of each mirror tile can be changed, and actually must be changed, since the accuracy of a manual fixation during the construction phase is not precise enough.

[1] The planned number of tiles changed in the past and might change again in the future. The current installation of the actuators assumes 877 mirrors.

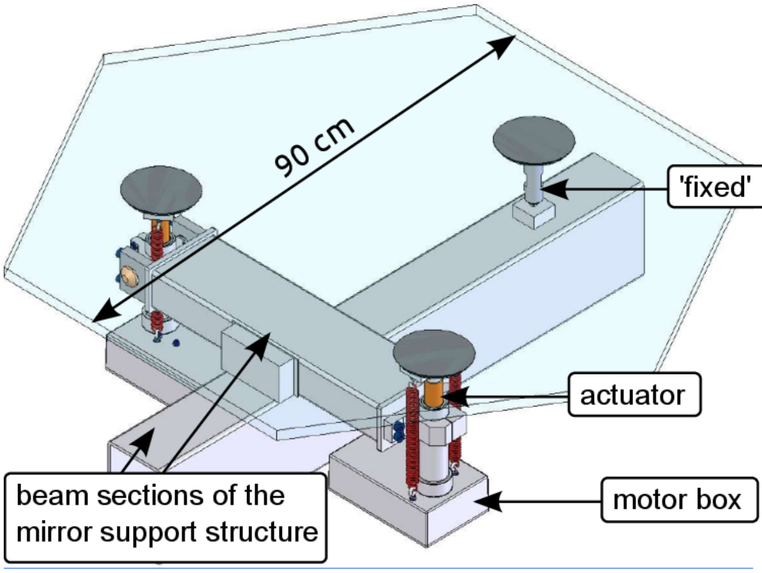


Figure 3.9: Model of a H.E.S.S. Phase 2 mirror and its support. One of the mounting points of the mirror is fixed (‘fixed’ because the length of the bolt is fixed, not its orientation). The other two points consist of motor driven actuators which can change the distance between the mirror and the supporting structure and thereby change the orientation of the mirror.

The mirror support structure is designed to be rigid enough to maintain stable imaging quality in zenith angle ranges between $0°$ and $45°$ (see Cornils et al. 2005). This results in the requirement for the mirror alignment system to operate whenever mirrors are exchanged and of course after the mirrors have been mounted for the first time. Nevertheless, the PSF of CT5 will be monitored regularly at the beginning and end of each observation month in order to be able to recognize any arising need to conduct intermediate realignments.

Within this thesis, parts of this mirror alignment system (precisely the control system) were developed. This comprised software as well as hardware development, which are the focus of the next chapters (Part I). In Part II I will finally turn to the analysis of the HESS source J1837-069.

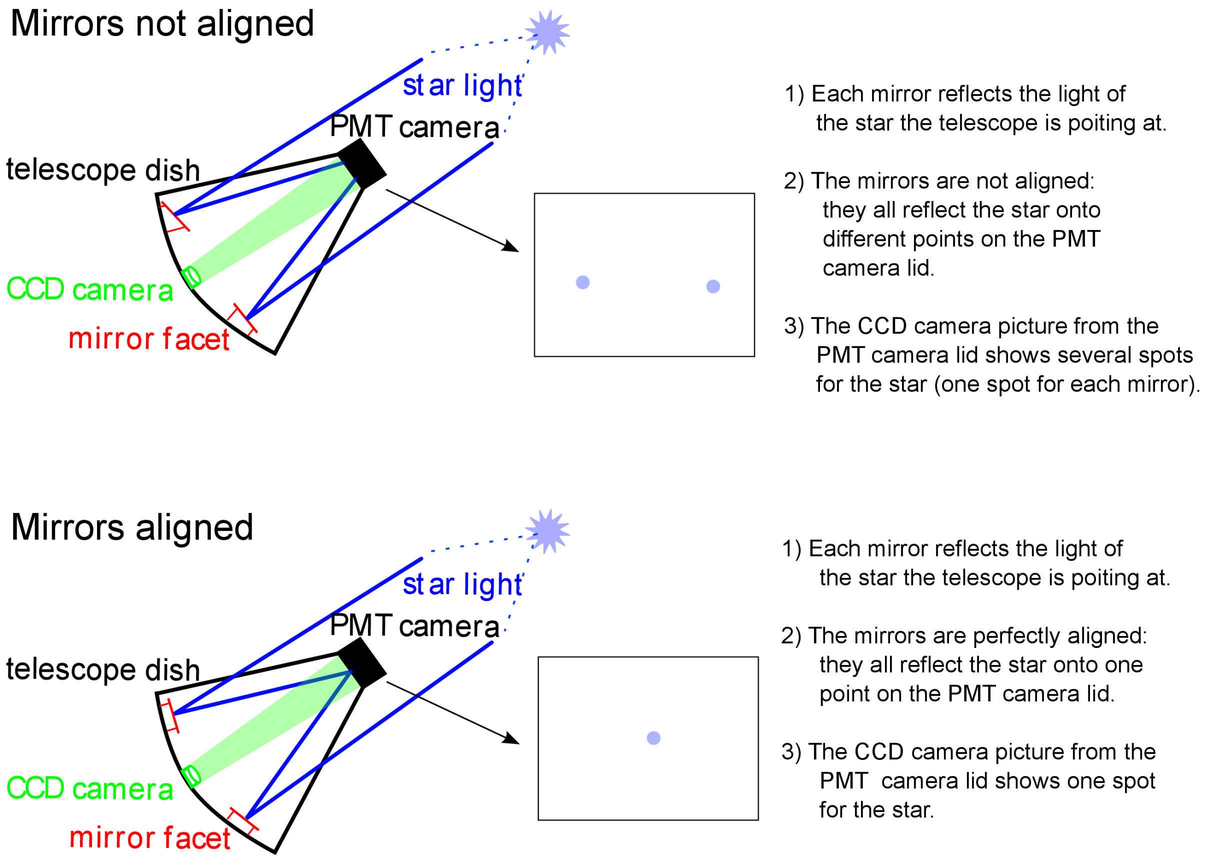
Part I

The Mirror Alignment Control System for H.E.S.S. phase II

Chapter 4

Introduction to MACS II

The alignment system for CT5 is based on the same principles as the alignment system of the first four telescopes. The system for the H.E.S.S. Phase 1 telescopes is described in detail in Cornils (2006).
The general alignment idea, reduced to its basics, is shown in Figure 4.1.



1) Each mirror reflects the light of the star the telescope is poiting at.

2) The mirrors are not aligned: they all reflect the star onto different points on the PMT camera lid.

3) The CCD camera picture from the PMT camera lid shows several spots for the star (one spot for each mirror).

1) Each mirror reflects the light of the star the telescope is poiting at.

2) The mirrors are perfectly aligned: they all reflect the star onto one point on the PMT camera lid.

3) The CCD camera picture from the PMT camera lid shows one spot for the star.

Figure 4.1: For the mirror alignment, the telescope is pointing at a star. The Lid-CCD camera takes pictures of the PMT camera lid, where the individual mirror facets reflect the light of the star. These reflections are called spots.

The telescope is pointing at a star which has to be bright and isolated. Each mirror facet reflects the star light onto the PMT camera lid and the Lid-CCD camera takes pictures of the reflected spots. If the telescope is not aligned, the spots created that way are spread out. While the simplified example in Figure 4.1 only shows two mirrors, CT5 will have about 900 mirrors. If these are not aligned properly, the picture will show up to 900 different spots for the same star. If the mirrors are aligned, they all focus the star light onto the same spot. The alignment system moves the actuators of the individual mirrors and takes pictures of the spot pattern in order to find out which spot belongs to which mirror. Once the alignment system has determined where each mirror is pointing at, it also has to find out how much each mirror has to be tilted through the actuators, so that all mirrors could reflect the star light onto the same position in the PMT camera lid. The idea to use stars to align the H.E.S.S. mirrors was first described in Hofmann (1998).

Each mirror is supported at three points (see Figure 3.9), two of which are motor driven actuators. Moving an actuator tilts the mirror. It is key to a successful alignment to understand how exactly the spot of a single mirror moves in the CCD image when one of the mirror actuators is moved by a defined range. Because of the fixed geometry of the actuators, two different spot positions and movements of a single actuator are sufficient to get a first order estimate of the transformation between actuator movement and spot movement. If this approximation based on the idealized geometry is not accurate enough, each actuator has to be moved. This results in four spot positions, which ideally surround the actual alignment position, in order to be able to interpolate the movement. If the four spot positions do not surround the alignment position, the spot movement has to extrapolated, which in itself is not ideal. Additionally, the approximation of a linear transformation between actuator movement and spot movement might no longer be valid.

In H.E.S.S. Phase 1, the described alignment procedure has been successfully used for the telescopes CT1 - CT4. But since CT5 will be considerably larger, the technical components and the technical setup of the H.E.S.S. Phase 1 alignment system could not be used for H.E.S.S. Phase 2. In H.E.S.S. Phase 1, the mirrors were addressed by a system of separate cables running from the electronic hut at the bottom of the telescope to each mirror. Applying the same system to over 900 mirrors would have resulted in a large amount of cables running from the ground into the dish. The cable chain would have become too large, putting lightning protection at risk. The problem here is that CT5 will not have the same lightning protection as the other telescopes. Setting up high enough external protection masts beside the telescope would have been very difficult. Thus, to minimize cabling and to take into account lightning protection, a new system had to be developed for CT5, called MACS II.

The technical components of the new alignment system and their development for CT5 are described in Chapter 5, while the automatic alignment algorithm and its implementation in software are described in Chapter 7.

A test rig showing the mounting of a mirror can be seen in Figure 4.2.[1] The mounting of

[1] At the time this thesis has been written, the CT5 telescope was not yet completed and therefore no pictures of the on site mounted mirrors can be shown.

the mirror on the test rig is done exactly as it will be on the H.E.S.S. Phase 2 telescope, but the mounting of the actuators is only an approximation of the real mirror support structure. The test rig was used to check the actuator movement accuracy, the speed and power consumption with the original mirrors weight and weight distribution in the testing phase of the hardware (see Chapter 6.5).

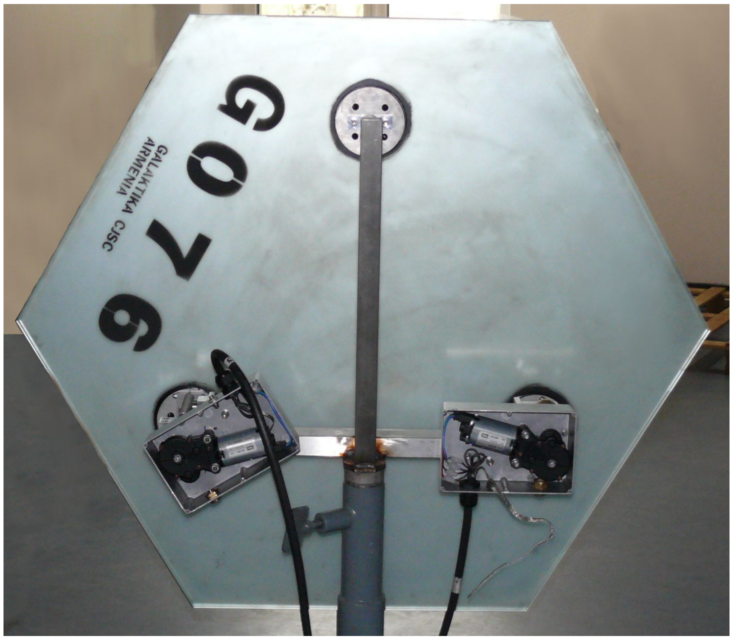


Figure 4.2: Test rig for a single mirror tile. The back of a H.E.S.S. Phase 2 mirror tile is affixed at three points.

The Mirror Alignment System for H.E.S.S. Phase 2 (MACS II) can be controlled by a H.E.S.S. staff member sitting in the control room. Via optical Ethernet, the operator can access the components in the central dish and also control the movement of each of the 42 mirrors in each of the 25 panels forming the dish (for an overview see Figure 4.3). It is possible to move only one or up to 25 mirrors at the same time. In the latter case, each of the 25 mirrors has to be situated in a different panel, since each panel control box can only control one mirror movement at the same time. This is a major improvement compared to the H.E.S.S. Phase 1 mirror alignment control system: In MACS I only one mirror could be moved at a time.

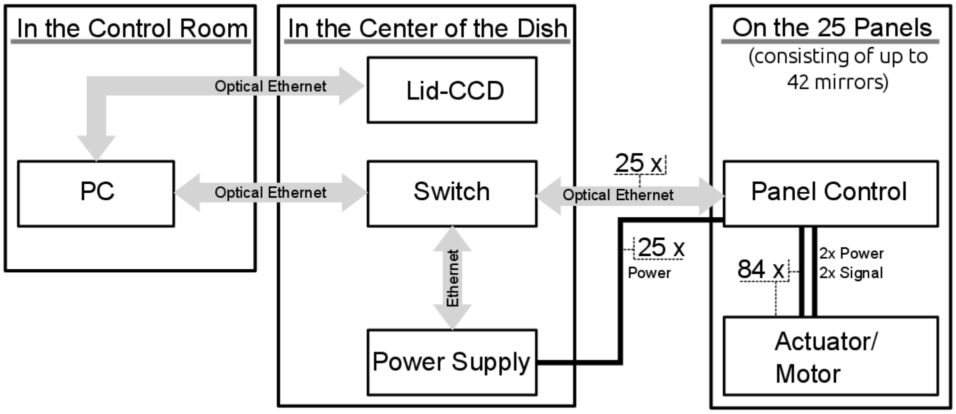


Figure 4.3: The main components of the CT5 mirror alignment system and their location.

The PC used in the control room will probably not be a dedicated PC, but one of the computers of the H.E.S.S. cluster as it needs no special hardware. A box with optical-to-Ethernet converters is placed in the center of the dish, a switch to route the 25 Ethernet components in the dish via a single optical Ethernet line to the network outsite the telescope, as well as the power supply. In each of the 25 panels, there are identical panel control boxes with optical-to-Ethernet converters, a System-on-a-chip (SoC) running uClinux and seven motor control boards. These will be connected to up to 84 motors, moving the 42 mirrors of each panel via the actuators mentioned. Together, these components in the control room, in the center of the dish and on the panels, constitute the MACS II.

Chapter 5

Components of the Alignment System

The mirror alignment system consists of different mechanical and electronics components, many developed as part of this research work. Most of the mechanical components were designed by the MPI-K in Heidelberg, whereas the electronic components were designed by the IAA Tübingen. The production of the mechanical components was done by a Polish company, the mass tests for quality assurance however were done by the IAA Tübingen and the precision measurement of the dimensional accuracy of a prototype was conducted by the MPI-K Heidelberg.
Additional hardware, part of or used by MACS II, consists of a PC in the control room, a power and switcher box in the center of the dish, and a CCD camera that takes pictures of the PMT camera lid.
Since the MACS II software poses no special requirements the alignment software will be installed on the computer cluster on the H.E.S.S. site. The software development was done on 32 bit and 64 bit Ubuntu systems and the production system will be a 64 bit Fedora system, but the Python code is written in a portable manner that would even allow the software to run on MacOS or Windows systems. The software development will be described in more detail in Chapter 7.
The power and switcher box contains a power supply, distributing the power to all the panel control boxes.
The CCD camera, used by MACS II, is mainly installed for pointing runs and/or other tasks that are independent of MACS II. The camera system for CT5 is not finalized.

5.1 Mechanical Components

The main mechanical components of MACS II are the actuators and the motors.
Each mirror is connected to the dish structure at three points. These connecting points consist of plates glued to the mirror backs. The glue itself is covered with a UV resisting paste. The plates are then screwed to the two actuators and the fixed connector. As already mentioned, the *length* of the fixed connector can not be changed (therefore it is normally refer to it as 'fixed'), but it still is *rotatable* in two ball joints, so its orientation is not fixed. One joint is common to all three connectors and accounts for the changing angle between the mirror and the connector, the other joint gives the 'fixed point' two degrees of freedom. Together with one of the other connectors which has a one degree of freedom joint, this takes the stress out of the mirrors, that would otherwise build up when the lengths of the actuators changes. The actuators, the fixed point and their mounting

have been developed at the Max-Planck-Institute in Heidelberg. All these components can be seen in Figure 5.1.

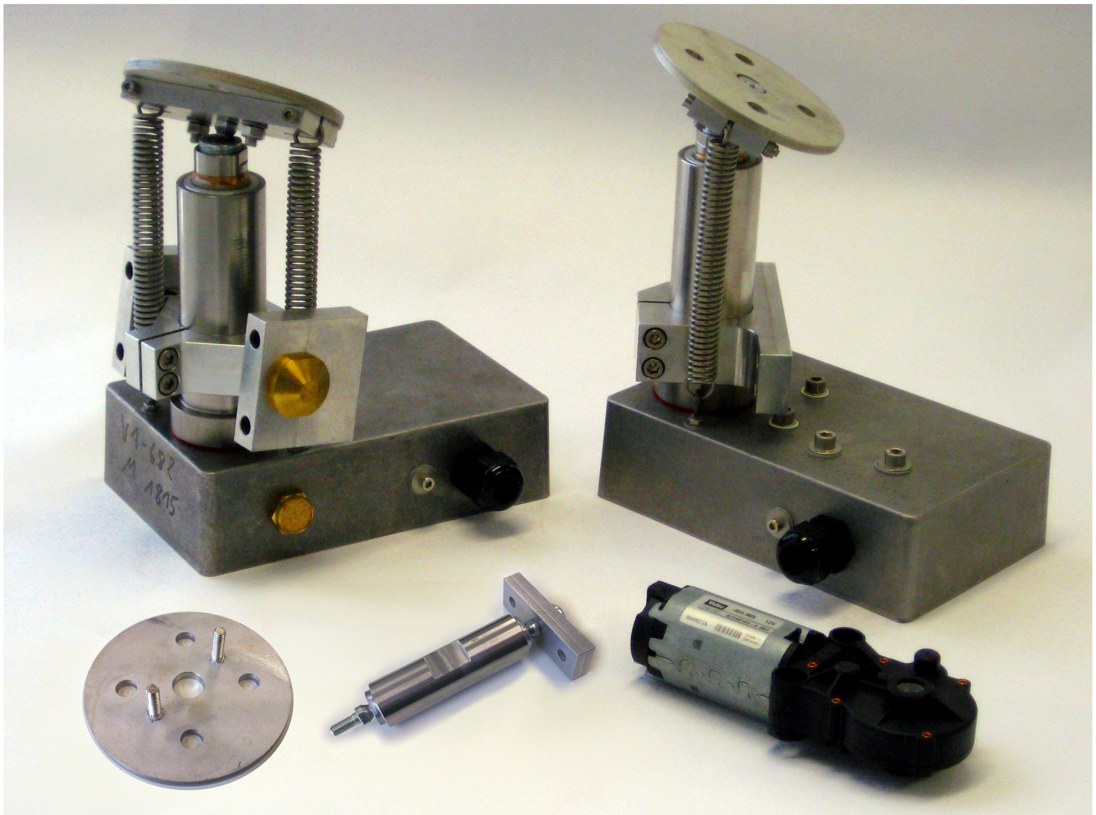


Figure 5.1: The mechanical components of the mirror alignment system: two actuators, a ‘fixed point’, a motor, and a plate that is glued to the back of the mirror. The actuators and fixations were developed at the MPI in Heidelberg by R. Kankanyan.

The spindle in the actuator has a pitch of 1 mm and a range of 40 mm. To minimize the clearance, the spindle is under constant pressure from two springs. The end positions of the spindle range are secured by disk springs. These disk springs give the control electronics enough time to stop the movement when the actuator is driven completely inwards or outwards, either because the time between two hallcounts (see below) becomes too long, or because the power consumption gets too high. In the current implementation of MACS II both stopping solutions are realized.

The motors are standard industry products developed by the company Valeo and are used in engine building and in automotive applications like seat back adjustments in cars. The chosen motor (GMPD 404 905) is a servo motor with an internal hallsensor close to the motor axis. The magnetic field for the hallsensor is created by a permanent magnet connected to the motor axis.

This hallsensor emits a signal, that can be used to keep track of the movement of the motor. One turning of the motor axis results in one impulse from the hallsensor, but since the motor includes a worm gear with a gear ratio of 210/1, one turning of the gear axis results in 210 impulses. These pulses have roughly square-wave reversals and therefore the raising and the falling edge of each signal can be used. Algorithms that detect both can have an overall resolution of 420 edges per revolution. The detection of one edge will be refered to as a *hallcount*. Hallcounts are the natural unit for actuator movements, just as pixels are the natural unit for mirror-spot movements in the Lid-CCD images.

Since the pitch of the thread in the actuator is 1 mm, we can control the length of the actuator to 2.3 µm. It must be noted however, that there is play in the mechanical components like the actuator, and there might also be deformations of the glued fixation points of the mirror or deformations of the mirror itself, which could decrease the accuracy of the alignment. Thus, the overall accuracy will be eventually estimated once the mirrors are mounted on the telescope.

The kind of motor movement detection described here can of course only be used to obtain information about the relative movement, not the absolute position of the actuator. In order to know the absolute position, it is necessary to store all movements in memory, or to use a counter, that sums all previous movements up. The latter is implemented as software on the Suzaku board (see Chapter 7) in conjunction with the main control software.

The operating voltage of the motors is 12 V. During no-load operation, the power consumption is typically between 600 mA and 800 mA.[1] With the springs connected, the consumption becomes 650 mA – 750 mA if the movement is in the direction of the spring force and 850 mA – 1000 mA against the direction of the spring force. On the telescope, the weight of the mirror adds as well. Since the telescope can be tilted, the weight is either perpendicular or parallel to the spring force. In those situations, the power consumption is between 650 mA – 1000 mA (perpendicular) and 680 mA – 1200 mA (parallel). The nominal torque of the motor is 2 Nm. Measurements by our group at the IAAT showed that the actuators required a torque of 0.65 Nm –- 0.7 Nm against the spring force and 0.35 Nm with the spring force. In the endpoints of the actuator, that are protected by disk springs, the torque increases rapidly over a range of 5 mm to 5 Nm. The maximal torque of the motor is dependent on the applied voltage. At 11.6 V, the maximal torque is 5.2 Nm and below that, the maximal torque drops rapidly until at 10.4 V, the motor does not move anymore. Above 11.6 V, the maximal torque is roughly constant (tested up to about 14 V).

In no-load operation, the nominal speed of the motor is 21 rpm. In the measurements where the motor was mounted to the actuator, the time for driving the complete actuator range (40 mm) was dependent on the applied voltage, with 130 sec (with) and 135 sec (against the spring force) at 13 V up to 163 sec (with) and 158 sec (against the spring force) at 11 V. With one rotation per 1 mm lift of the actuator, the speed is between 18.5 rpm and 14.7 rpm. Translated in the relevant unit for MACS II this is about 0.001 s/hallcount. The time between two hallcounts is of course not constant. It is higher during starting and

[1] Measurements were conducted with two actuators and a simple ammeter with a reading accuracy of ±10 mA.

Table 5.1: Some useful numbers of the mirror alignment system. Starred (⋆) entries are typical values for rule of thumb calculations.

Description	Value
Hallsensor pulses per turn	210
Hallsensor pulse edges per turn	420
Lift per turning of the motor	1 mm
Lift per hallsensor edge	2.3 µm
Lever arm of actuator movement	450 mm
Mirror tilt per hallsensor edge	1.1 arcsec or 0.005 mrad
Light spot movement per hallsensor edge	2.2 arcsec or 0.01 mrad
Pixel size of the PMT-camera	1.22 mrad
Time between two hallsensor edges	1/120 s ⋆
Actuator range	4 cm ⋆
Time used to drive the actuator range	140 s ⋆

stopping of the motor. A list of ballpark figures of the alignment system is given in Table 5.1.

5.2 Electronic Components

The custom made electronic components were designed at the IAAT. They consist of a motor control board, a panel control board and a bus system that connects those components. Together with a commercially available System-on-a-chip on the panel control board and an optical-to-Ethernet converter, these components form a panel control box (see Figure 5.2 and Figure 6.9).

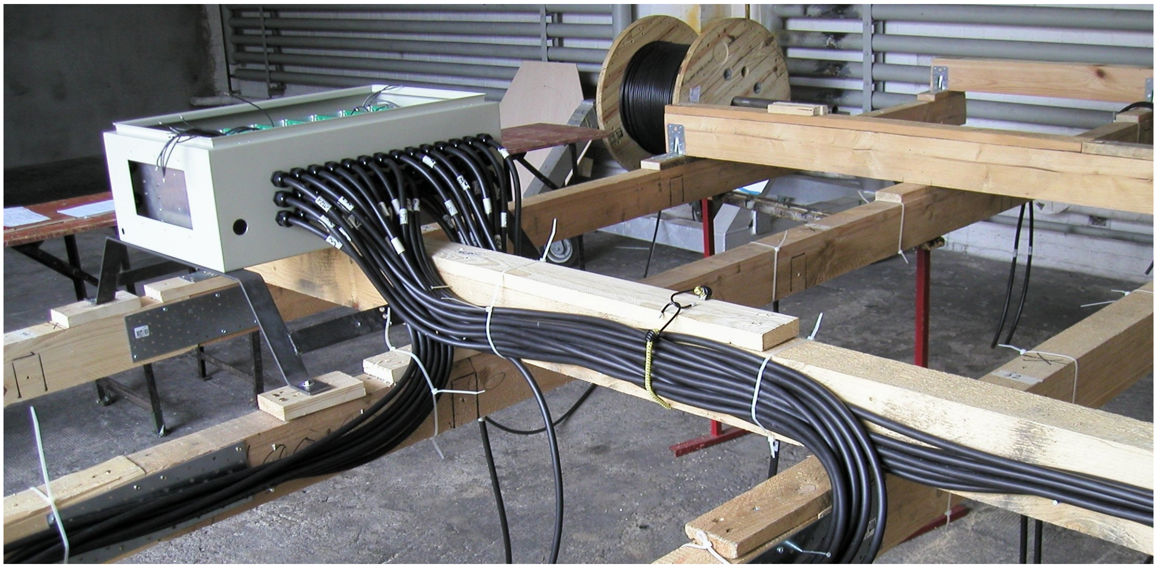


Figure 5.2: Close to the panel control boxes, the cable routing becomes more difficult and was therefore tested on a model of a part of a panel that was build at the IAAT.

5.2.1 The Motor Board

The main components of the motor board are the motor drivers, a multiplexer, and a demultiplexer.

The motor drivers are H-bridges (LMD18200T), special electronic circuits that are used to control DC motors by providing a way to select the direction the motor turns and by braking the motor. The motor drivers also include overpower protection, to protect against induced currents from suddenly stuck motors. In that case the device will cut the power off. When the motor power is down, the motor driver will periodically try to turn the power back on, resulting in a on and off cycle. When this happens, the main control (see Chapter 6.1) has time enough to permanently shut the power off.

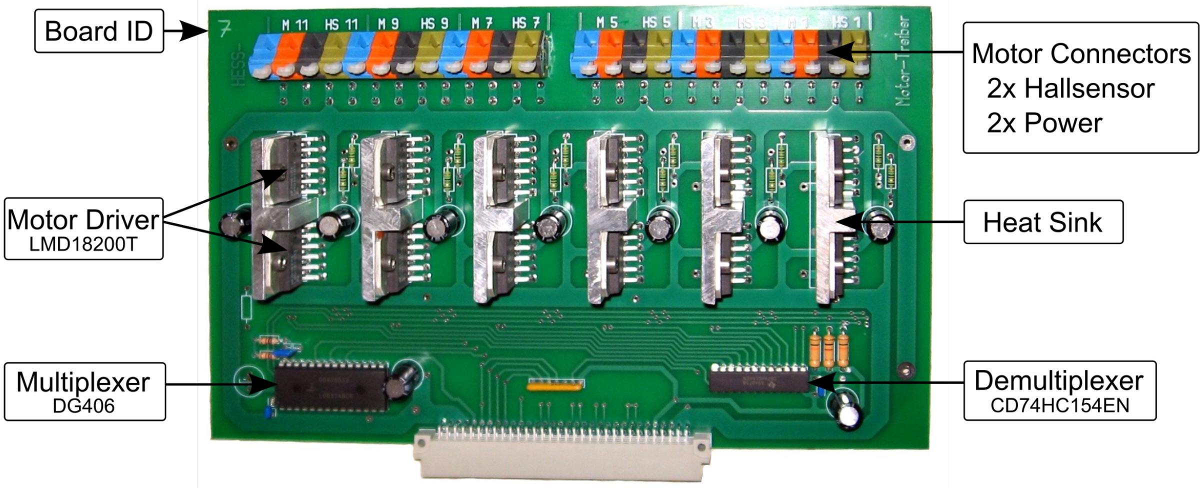


Figure 5.3: The motor board, designed by S. Vetter at the IAAT, is responsible for decoding motor selection from the panel control board as well as driving the motors via dedicated ICs.

The bus in the panel control box runs a power line and four signal lines from each motor board. The demultiplexer (CD74HC154EN) on the motor board is then used to select the motor that is powered by the power line from the bus.

The bus runs another five signal lines, four to select the hallsensor output of the motors via the multiplexer (DG406DZJ) and one for the selected hallsignal.

5.2.2 The Panel Control Board

From early on in the design process of MACS II it was clear that the communication mechanism should be via Ethernet. To supply the communication mechanism is the purpose of the panel control board. While Ethernet is standardized and fast, it is not fast enough to allow reliable real time supervision of many moving motors[1]. For this reason,

[1] The maximum possible Ethernet data-rate would be fast enough of course, but cheap, small, embedded computers providing Ethernet communication are typically not fast enough.

the control board does not only provide Ethernet access but performs the real-time supervision of the motors. The board has therefore to interface the high-level alignment software that runs on the control PC and the electronical and mechanical hardware.

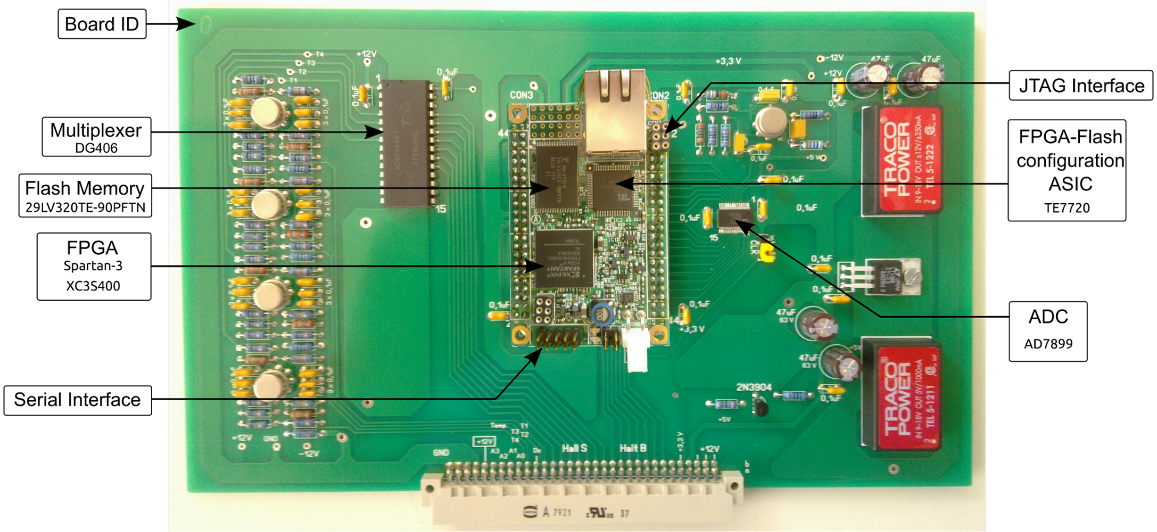


Figure 5.4: The panel control board was designed by S. Vetter at the IAAT. It contains among others a multiplexer, an ADC, and the Suzaku board. The Suzaku board is a commercial component from Atmark-Techno[1], that was expanded by custom hardware and software for MACS II.

The main components on the panel control board are a multiplexer, an ADC, and the Suzaku board.

The multiplexer is used to send input to the ADC, which is then read out by the Suzaku board. The input for the multiplexer is coming from the seven motor boards, four (optional) temperature sensors, and a Hall effect-based linear current sensor (ACS704) that is located on the power line of the bus. Temperature sensors (PT100) were planned at an earlier stage and the connectors for the sensors are in the panel control box. In the current status of implementation no temperature sensors are installed. However the board has been constructed also to readout temperature sensors.

The linear current sensor is used to prevent over currents due to motor failures, stuck actuators or actuators driven into the disk springs at the end of their range.

The other inputs to the multiplexer are the hallsensor outputs from the motors, first multiplexed on the motor boards, to only select the right motor, then multiplexed on the control board to select the right board.

All these components are controlled by the FPGA hardware (Field programmable gate array, see 6.1) and software on the Suzaku board. The development and testing of the hardware and software on the Suzaku board was a substantial part of this thesis and will be described in the next chapters. We wish to stress that the development for the Suzaku board was challenging and demanding. The hardware development process and some of the exhibited challenges will be described before turning to the software part of the thesis in Chapter 7.

Chapter 6

Hardware Development for MACS II

The Suzaku board on the panel control board is a commercially available embedded solution.[1] It provides the basis for a diverse range of applications, from data monitoring, over motor control (like in the case of MACS II), to remote video operations. The reason for choosing a solution like the Suzaku board is the need to make devices available over the network that are normally hard to connect to standard CPUs via e.g. USB.
While this is the range of possibilities Suzaku boards offer, this doesn't mean, that these possibilites are already implemented on the board, ready to be used. The uClinux kernel running on the Suzaku board already provides a full network stack. But other applications as well as connections to devices outside the board still have to be implemented by the user either as hardware extensions to the FPGA or as software extensions running on the uClinux kernel.
All real-time extensions or computing intensive operations have to be implemented in hardware, because the CPU in the Suzaku board is too slow for software solutions.
For H.E.S.S., the outside devices which had to be connected to the Suzaku board were the actuators moving the mirrors. Thus, for MACS II, we had to develop the connection between these and the Suzaku board via the components in the panel control board. For the real-time observation of the motors, additional hardware components were introduced to the FPGA on the Suzaku board.
The following chapters describe FPGAs and the FPGA development process in more detail. Then the focus will be on the software part of MACS II.

6.1 Field Programmable Gate Arrays (FPGAs)

Implementation of hardware designs that require large amounts of logic gates can be done (amongst others) with Masked Programmed Gate Arrays (MPGA) or Programmable Logic Devices (PLD). An MPGA consists of predefined transistors with a fixed wiring for each design. Wiring the transistors is done in the manufacturing process via specially designed masks for each logic implementation. Since every new design requires a new mask, this increases the development costs. It also increases the time between the design and the finished device. This disadvantage was addressed in 1984 by Xilinx (Paulin 2009), the company which first created Field Programmable Gate Arrays (FPGA), which contain programmable logic cells and also programmable wiring (so called interconnects). The

[1] Atmark Techno: `http://suzaku-en.atmark-techno.com/`

'Field' part of the name FPGA comes from the idea that this can be done 'in the field' as opposed to 'by a manufacturer in the lab'.

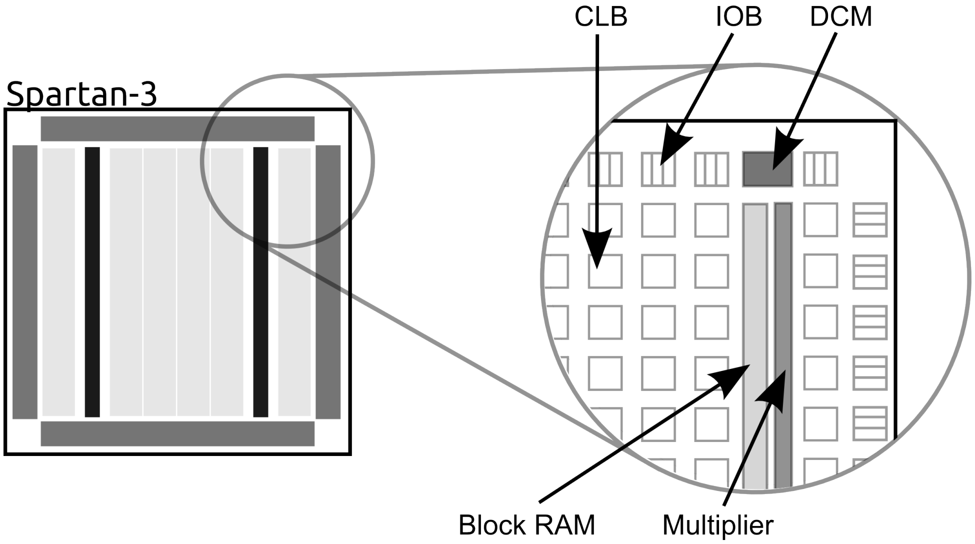


Figure 6.1: The Spartan3 architecture overview. The building blocks are: Configurable Logic Blocks (CLB), Input/Output Blocks (IOB), Block RAM, Multiplier Blocks, and Digital Clock Managers (DCM) (see Xilinx 2009).

The Suzaku board includes a Spartan3 FPGA from Xilinx (XC3S400). This is a rather cheap and small FPGA compared to other current Xilinx flagship FPGAs like the Virtex-7. The Spartan3 contains Configurable Logic Blocks (CLBs), Input/Output Blocks (IOBs), Block-RAM, multiplier blocks and Digital Clock Managers (DCM) (see Xilinx 2009). The general design can be seen in Figure 6.1.

The CLBs are RAM-based look-up tables that can be used as flip-flops and storage elements to perform various logical functions. The data flow between these CLBs is controlled by switch matrices as shown in Figure 6.2.

The IOBs control the data flow between the I/O pins and the internal logic of the FPGA. The IOBs are located on the outside of the chip, therefore the number of IOBs scales linearly with the size of the chip, unlike the number of CLBs, which scales quadratically.
The multiplier blocks take two 18 bit binary numbers and provide the product. The Digital Clock Managers are used to synchronize the clock signal across the whole FPGA by distributing, delaying, dividing, and shifting the clock signals. In more complex designs, multiple clock signals are possible for different regions in the FPGA. But smaller designs like the MACS II hardware configuration rely on a single synchronous clock signal.
The advantages of FPGAs over fixed layout devices seem obvious: faster design-test cycles and lower development costs since errors in the design can be undone. The disadvantages over MPGAs and other alternatives for logic design can easily be explained: the flexibility requires additional components and therefore decreases the density of logic blocks and increases the per-chip cost. On the other hand, if a standard CPU with assembler or compiled C code can be used, the development time and complexity, as well

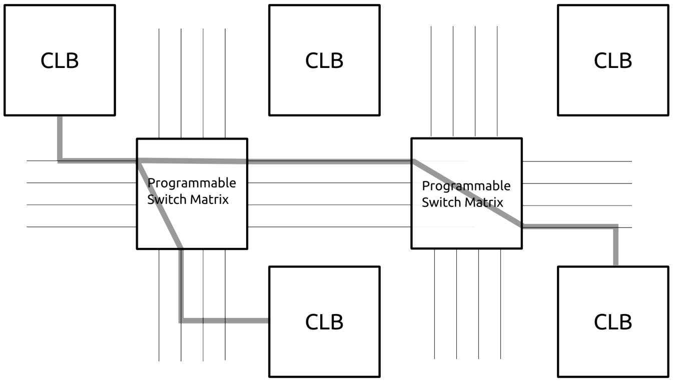


Figure 6.2: Schematic view of the result of a routing step. Programmable switch matrices connect the inputs and outputs of CLBs. The more processes make use of a single signal in the VHDL description, the more complicated the routing will be and the longer the routes have to be, in order to connect CLBs that are far apart. This in turn leads to longer signal distribution times and a slower overall performance of the design.

as the cost, can be lowered considerably. This is only possible though, when no parallel processing is required and standard connections between the application devices and the CPU can be used.

6.2 *Introduction to Hardware Design with FPGAs*

In a typical FPGA design process, the desired functionality is first described in a Hardware Description Language (HDL) similar to the description of an Application Specific Integrated Circuit (ASIC). Also in terms of the logic that can be implemented, there is no difference between an FPGA and an ASIC. The two most popular hardware description languages are VHDL (Very High Speed Integrated Circuit Hardware Description Language) and Verilog. Both languages are rather high-level languages with many libraries for e.g. mathematical operations. In the case of MACS II development, the decision as to which language should be used was already made by the designers of the Suzaku board, because large parts of the Suzaku hardware are implemented in VHDL and included as IP cores. ‘IP core’ stands for ‘Intellectual Property core’ and describes a unit of logic that can be *reused* by others, often including licensing. *Changing* IP cores however is not possible.

6.2.1 The FPGA Design Process

The FPGA design process includes several steps: When the VHDL description is written, it is translated to a netlist. Then it must be fit to the actual hardware by a process step called place-and-route. The result can then in principle be simulated by a separate software and tested, before it is transferred to the FPGA. This process of routing is shown in Figure 6.3.

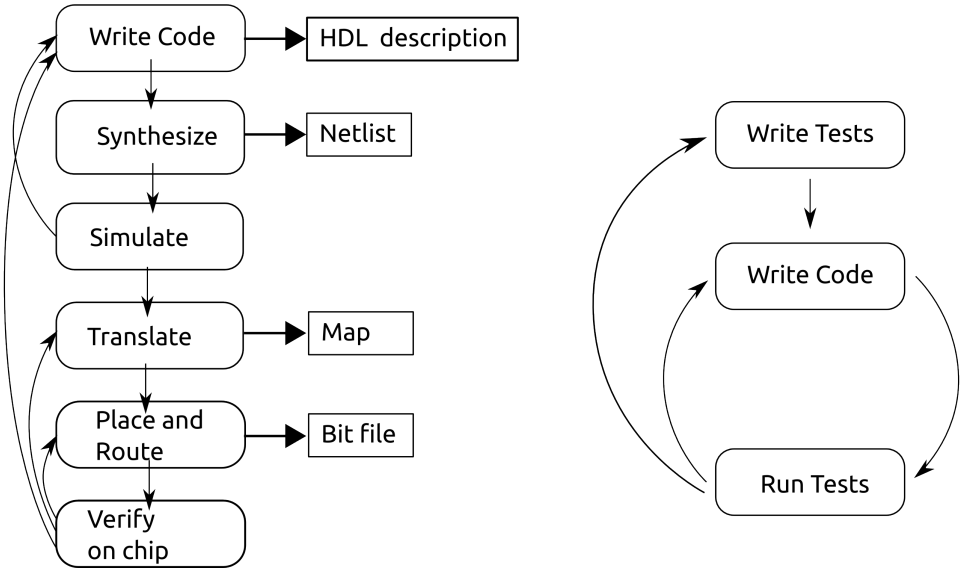


Figure 6.3: A typical FPGA hardware design process *(left)* includes a simulation step and many intermediate steps as well as different products created in each step. Compared to this a typical test-driven software development process in programming languages with an interpreter *(right)* seems less complex. Both processes were used to develop MACS II, which has hardware as well as software parts.

When comparing hardware development with software development, there appear not only different language concepts, but there are also differences in the design process. One of the biggest problems is the time it takes between implementing (changing) the design and being able to see the result. In current software development, with the rise of interpreted languages, the time between changing and testing a code is in the order of seconds to minutes. Even with compiling and linking, the typical times are a few minutes. In the FPGA design setup that was used for implementing the mirror alignment logic, the typical time was in the order of 1h, as usual in hardware development. This has of course an impact on the way how hardware is developed. Precisely this means that a simulation step is necessary. It can be used to check the functional integrity of the design, and thus becomes an important tool to reduce the waiting time. If functional errors occur within the simulation step, they can be fixed without transferring the design onto actual hardware. Unfortunately, functional integrity can not *guarantee* that the design does actually work on the hardware, since in the simulation many of the components are idealized.

Since the field of FPGA programming is rather new compared to classical software development, the tools are also not en-par with the state-of-the-art software development debugging and testing tools. During the design of the mirror alignment system, which was done with Xilinx Embedded Development Kit (EDK) and Xilinx Platform Studio 7.1, many flaws of the design software appeared. When starting with the MACS II project, we were convinced by the many advantages the Suzaku board offered to our project.

Unfortunately, programmers also have to deal with the bugs hidden in external software. Two serious and time consuming bugs occurred which will be described shortly:

- Multiple assignments of a signal in a process were not caught by the software, but they were synthesized, mapped, and placed&routed. This caused undefined behaviour of the final design. In software development, these errors are typically already captured during writing by common editor macros and syntax highlighting routines.
- If the place&route process could not meet the timing requirements of the design, neither a warning nor an error was shown, but again, all further steps until the final design were completed. This again resulted in undefined behaviour of the final design bit file.

Although this might sound harmless, it is very difficult to debug from the final FPGA behaviour, because 'undefined' behaviour often means the correct behaviour. It is also difficult to distinguish between these bugs and e.g. loose connections on the circuit boards. And the latter also occurred during the MACS II development. Especially the second bug in the design software caused major delays in the development, since it was found very late. Major parts of the FPGA design needed to be rewritten during the last months of the dissertation project to fix it. Xilinx obviously realized the problem later on on their own, because the bugs have been fixed in later versions of the Xilinx software Platform Studio.

6.2.2 A Comparison Between FPGA Programming and Software Development

The VHDL syntax was inspired by Ada and Pascal. Nevertheless, the design process requires some rethinking, when coming from a software development background. The biggest difference is the control flow in VHDL compared to software languages. VHDL is called a *dataflow language* compared to a *procedural language*. VHDL code is organized into one entity per hardware module. Each entity is then divided into entity declaration, architecture, configuration, package declaration, and package body. The package body can further describe hardware in the way of Dataflow Modeling, Behavioral Modeling, or Structural Modeling. If using Behavioral Modeling, the package body contains processes, signal assignments, and sequential assignments. Processes bundle lines of code that are run sequential, although assignments within a process are only finished at the beginning of the next process run. This difference between VHDL and e.g. Python code with regard to in-process assignments can be seen in Listing 6.1.

Listing 6.1: VHDL code sample1

```
A <= 1;
if A = 1 then
   B <= 0;
else
   B <= 1;
endif;
```

Listing 6.2: Python comparison sample 1

```
A = 1
if A == 1:
   B = 0
else:
   B = 1
```

This example shows two integer signals A and B and a conditional statement in a single process. In the VHDL code when the lines are 'executed'[1], the value of B at the end of the cycle can not be determined from these lines alone. In contrary the Python code while looking similar is fully determined by the given lines in Listing 6.2. The VHDL behaviour is different, because the conditional statement uses the value of `A` from the last cycle. That means both `A <= 1` and `if A = 1 then` are 'executed' at the same time and the result of `A <= 1` can therefore not influence the outcome of `if A = 1 then` in the same cycle.

Apart from the effect of in-process assignments, there is also another big difference in the behaviour between procedural languages and dataflow languages like VHDL. In VHDL, all processes[2] are 'executed' at the same time. The consequence of this is that having a complicated assignment in one process increases the time that every process takes: this affects the process with the complicated assignment as well as all other processes in the design. Again, this is caused by the fact that a new cycle can only start, when all processes have finished all commands. It is a good practise to keep all processes as simple as possible and try to avoid complex commands like multiplication and division (except for division by a power of 2, since this can be expressed as a shift operation).

Yet another difference compared to best practices in software development is the duplication of signals. Given signals `A` and `B`, where the state of `A` is accessed in several processes, code like the one in Listing 6.3 can often be seen.

Listing 6.3: VHDL code sample2

```
A2 <= A;
if A2 = 1 then
   B <= 0;
else
   B <= 1;
endif;
```

Listing 6.4: Python comparison sample 2

```
A2 = A
if A2 == 1:
   B = 0
else:
   B = 1
```

[1] When VHDL is synthesised and loaded into an FPGA, the VHDL code is not 'executed' on a processor in a way software code is normally executed. As 'executed' is the common term used for describing what happens when code becomes active, I will stick to the term. But strictly speaking what happens in VHDL code is that the code configures the interconnects and logic blocks of the FPGA.

[2] If the design is written synchronous, which it typically is. In the case of the user logic of MACS II, that is the case.

In this example, the difference to a simple `if A = 1 then` is, that the place&route process only has to route the signal `A` up to a storage gate and then another signal `A2` to the CLB that does the comparison. Therefore the overall time for this process can potentially be reduced and the routing can be easier. There is of course the additional difference that the information now takes an additional cycle to be processed by the `if else` construct. The same code lines in Python (Listing 6.4) would be considered useless (at best) in software development, since they would either increase execution time and memory consumption or would be completely removed by the interpreter or compiler.

6.3 The Suzaku Board

The Suzaku board is best described as a System-on-a-chip (SoC). This is a rather vague name, basically meaning that all components that are typically on a full computer are build onto a single chip. Normally, larger SoCs are not completely on a single chip, but are tightly integrated systems with external memory chips. On the Suzaku board the main component is the Spartan3 FPGA. Other external components are two memory chips, a 16MB DRAM memory and a 4MB Flash memory, a crystal oscillator (3.6864 MHz), a LAN controller, a custom FPGA and Flash configuration IC from Tokyo Electron Device Limited (TE7720), a RE232C receiver, and an LED. The main components of the Suzaku board can be seen in Figure 6.4.

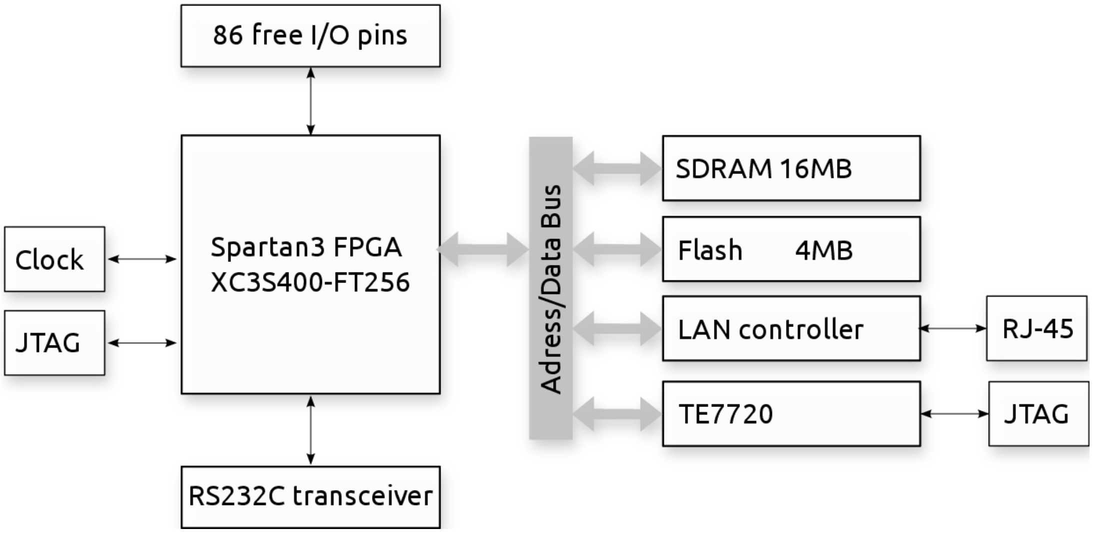


Figure 6.4: The main components of the Suzaku board.

Most components that are not in the FPGA are connectors, which qualifies for the Suzaku to be called a System-on-a-chip. The external bus (external means here outside the FPGA) connects the external memory and connectors. Apart from the external bus, there are two more internal buses, that connect the FPGA components. The Flash memory is used to store the FPGA bit file as well as the Linux kernel and all additional software.
Communication with the board can either be done via serial console (via the UART lite

IP Core), via Ethernet or via JTAG. The two JTAG interfaces on the board are different from each other, one allowing access to the FPGA for programming (which does only last until power is shut down), the other can be used only together with the Little Bear Player 2 (lbplayer2) via the TE7720.
The internal layout of the FPGA can be seen in Figure 6.5.

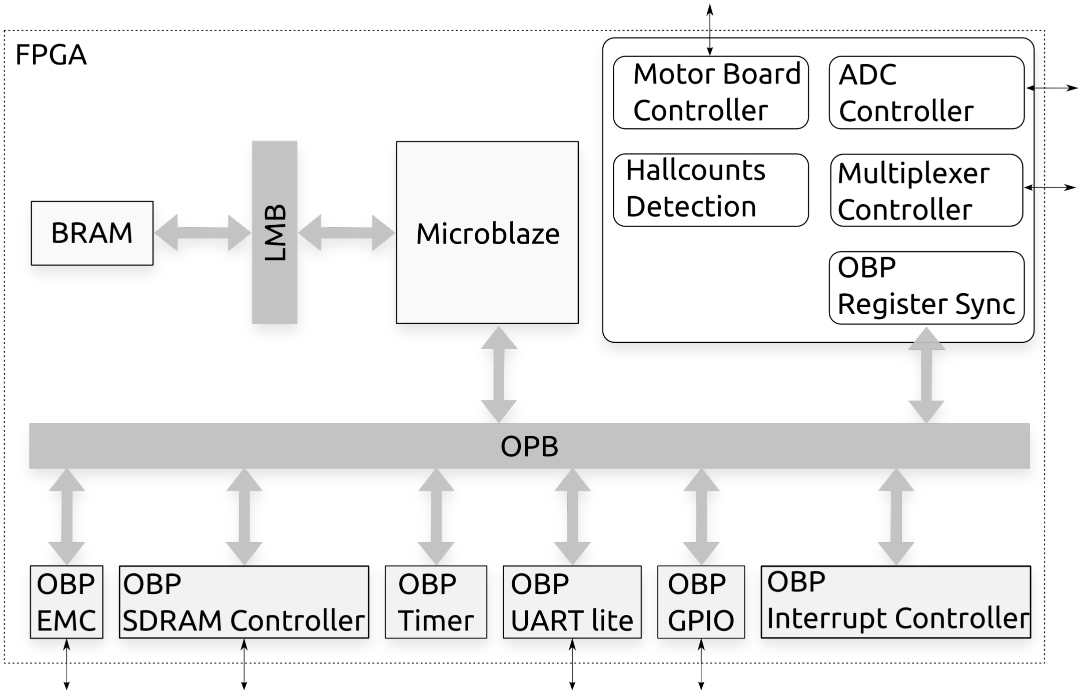


Figure 6.5: Internal components of the FPGA on the Suzaku board. Grey components are IP cores from Xilinx. Components with rounded corners were developed during to this thesis. The narrow black arrows show the connections to the external components of the Suzaku board.

6.3.1 The Onchip Peripheral Bus (OPB)

The Onchip Peripheral Bus (originally designed by IBM, see IBM Corporation (2001)), provides the link between the CPU architecture and the other architectures. The OPB is a 32 bit data and 32 bit address bus. It is a fully synchronous single clock edge bus, with a single cycle transfer time between OPB master and OPB slave. The implementation from Xilinx does not implement the dynamic resizing feature of the original IBM design. This is the bus that can also be used for added user logic.[1] Already included in the Suzaku distribution is the OPB EMC (OPB external memory controller), an IP core from Xilinx that is used to access the Flash and SDRAM memory chips. Other IP cores are also included to control various other components on the Suzaku board. Note that the register synchronization speed between the user logic and the software running on

[1] *User logic* is the term that is used in all the Suzaku manuals to describe FPGA components, IP cores, that are not already present in the Suzaku board, but are added later by *users* of Suzaku.

Table 6.1: Byte and bit order used in the MACS II components (shown here is the layout of a 32 bit word). MSBit/MSByte stands for *most significant bit / byte*. LS accordingly stands for *least significant*.

Big-Endian bit-reversed				
Byte address	n	n+1	n+2	n+3
Byte sign.	MSByte			LSByte
Bit sign.	MSBit -			- LSBit
Big-Endian				
Byte address	n	n+1	n+2	n+3
Byte sign.	MSByte			LSByte
Bit sign.	LSBit - MSBit	LSBit - MSBit	LSBit - MSBit	LSBit - MSBit
Little-Endian				
Byte address	n	n+1	n+2	n+3
Byte sign.	LSByte			MSByte
Bit sign.	MSBit - LSBit	MSBit - LSBit	MSBit - LSBit	MSBit - LSBit

the Microblaze CPU is relatively low. This arises from the fact that many components are connected through this bus.

Each OPB slave must implement the address decoding logic, the OPB slave interface logic, and application specific logic. The address decoding logic is the most complex single cycle logic that was implemented in the user logic for MACS II. Therefore it was ultimately the reason, why the design had to be redone at the end of the thesis, because the Xilinx software was not able to place and route the netlist of that process within the timing requirements of the other FPGA components. As explained earlier, this was not reported as any kind of error from the Xilinx software.

6.3.2 The Microblaze CPU

The processor on the Suzaku board is a Microblaze IP core from Xilinx, a 32 bit reduced instruction set (RISC) Harvard architecture. It is highly configurable, due to its nature as a soft core rather than a real chip. The fixed feature set that every Microblaze implements consists of 32 general purpose registers of 32 bit, a 32 bit instruction word with three operands and two addressing modes, a 32 bit address bus and a single issue pipeline. Among the optional features that are enabled in the Suzaku version are the OPB controllers and LMB (local memory bus) controllers, but for example not the floating point conversion and square root instructions or the Memory Management Unit (MMU). The absence of a MMU makes a big difference to the software development and is described in the software chapter later on (see Chapter 7.1).

The Microblaze, unlike current Intel architectures, implements a Big-Endian format, a format that is common in e.g PowerPCs. But the Microblaze additionally has a bit-reversed format, resulting in the first register scheme shown in Table 6.1.

The data type layout of 32 bit words is important, since the communication between the software running on the Microblaze and the user logic in the FPGA via the OPB synchronises several 32 bit registers. While the software can interpret these registers as custom data types, they can also directly be treated as e.g. integers. The user logic in the FPGA must therefore take care to align the bits in the right order. On the other end of the communication, within the transfer of information via TCP/IP with the control PC, the byte order used is the network byte order, which is defined as Big-Endian. The software on the PC outside the telescope will probably be running on Intel hardware which is Little-Endian. Together the chain therefore goes from custom 1 bit to 32 bit registers, via Big-Endian bit-reversed over Big-Endian to Little-Endian. To reduce this potential source of conversion errors, the custom registers in the FPGA design are kept to 32 bit Big-Endian bit-reversed just like the processor data type layout and the data send via TCP/IP are send as 1 byte ASCII characters only. The second is possible, since all time critical operations are done directly in the user logic. Only reports and commands are send via Ethernet from the control PC.

6.3.3 Booting and Programming the Suzaku Board

Booting the Suzaku board to full uClinux (see Chapter 7.1) takes around 23 seconds.

During this time several steps are processed: firstly, the first stage boot loader BBoot is started, which can then secondly either be used to recreate the second stage boot loader, if the boot mode jumper is set, or otherwise start the second stage boot loader *hermit*. BBoot is placed in the BRAM part of the FPGA and should in principle be able to rescue the Suzaku as long as the FPGA starts up. The mode that can be used to overwrite the second stage boot loader is called Motorola-S Mode (after the file format used to transfer the second stage boot loader). The second stage boot loader is stored in the Flash memory. Thirdly, in the Boot Loader Mode, the second stage boot loader can be either used to overwrite the hardware configuration or the uClinux image, which both reside in Flash memory as well. In the default mode, the Auto Boot Mode, the second stage boot loader loads the FPGA configuration from the flash memory into the FPGA and the uClinux image into DRAM and boots uClinux. Since the Ethernet controller and the TCP/IP stack are part of the FPGA configuration or uClinux, they can not be used to change the second stage boot loader or the other configuration components.

The *netflash* utility described in the Suzaku-S software manual does not work with the Suzaku-S version used in MACS II, but could do so in the future. To make it work in the future, the Suzaku-S board had to be further developed, so that the netflash utility could be used to reprogram the hardware as well as the software part during run-time via Ethernet (this would still require a re-boot).

The standard JTAG interface can not be used to program the FPGA permanently, since the programming is volatile and must always be stored in a permanent memory like flash, but the standard JTAG interface makes only the Spartan3 accessible.

Table 6.2: Software - hardware synchronised registers

Register	Read	Write	Integer	Multivalue
Actual hallcounts	✓		✓	
Required hallcounts	✓	✓	✓	
Power threshold	✓	✓	✓	
Hallsensor high state	✓	✓	✓	
Hallsensor low state	✓	✓	✓	
Hallsensor state trigger	✓	✓	✓	
Start timeout	✓	✓	✓	
Main timeout	✓	✓	✓	
Motor board direction	✓	✓		✓
Temperature	✓			✓
Power	✓		✓	
Status	✓			✓
Debug	✓			✓
Command	✓	✓		✓

6.4 *Hardware Implementation*

The user logic that was added to the FPGA design can be divided into different parts, each is implemented as one or more separate process(es): the OPB synchronization, the ADC controller, the multiplexer controller, the hallcount detection, and the motor board controller.[1]

6.4.1 OPB Register Synchronization

The purpose of these processes is to control the synchronization of registers between the CPU and the VHDL design. The initial plan was to synchronize all registers both ways, but this turned out to be too complex for the FPGA (and the synthesis tools from Xilinx). So, in the final implementation, part of the registers are read-only, others are read-writable. Most registers contain integer values and are therefore completely ‘filled’ by a single number. Other registers contain several distinct bits of information (in the literal sense) and must be decomposed on the software side. See Table 6.2 for a list of registers.

6.4.2 ADC Controller and Multiplexer Controller

The ADC on the Suzaku board is controlled by several processes in the FPGA. Although not necessary from the wiring of the board, this also includes the ADC clock. This clock

[1] The complete project – the suzaku-s files including the newly added components as well as the software components which will be described later – are available to IAAT members via a git repository at: `git@aithp1.ait.physik.uni-tuebingen.de:thesis/MACSII.git`.

was provided, because it makes the behaviour of the combined system more stable if only one clock is used. Otherwise, a frequency shift could lead to unexpected results, when reading from the ADC output.

The multiplexer on the Suzaku board is used to switch the input to the ADC from the hallsensors, the current sensor, and the temperature sensors. The temperature sensors are currently only readable when no motor is running, and only when explicitly requested. The main reason for this is the complexity of the switching process and the register synchronization. Both had to be reduced in order to help the VHDL synthesizer to reduce routing delay times to meet the requirements of the Xilinx IP cores. Initial designs included temperature thresholds for the different temperature sensors. This is no longer available in the current implementation.

6.4.3 Hallcount Detection

Hallcounts are defined in this context as the edges of the hallsensor signal. The basic idea is to identify the moments, when the signal switches from a high state to a low state or the other way around. This switch can not be detected by simply having a 'last value' and 'this value' register and by comparing both registers to each other, because the hallsensor output can have spikes, which can, especially at the beginning of the motor movement, lead to a miscounting of the edges.

Two different methods are used to improve the situation. The first is a running mean implementation which replaces the original signal by the mean of the last eight ADC conversions. The second is the introduction of a minimal amount of times the signal has to be above a high threshold or below a low threshold in order to be recognized as a valid high state or low state.

We have extensively tested the hallcount detection mechanism. For small numbers of counts, this was done by comparing the output of a oscillograph manually with the number of counts registered by the hardware. This could be done to up to ~ 500 counts and in all of these tests the hardware detection was absolute reliable.

As mentioned before, the starts are especially prone to produce errors in the counting process. Therefore additional long running tests were conducted, where the start position and the end position of the actuator was measured by a dial gauge that was fixed to the mirror test rig (see Figure 6.7). The test consisted of many (several thousand) movements of the actuator in one direction and the same distance in two separate movements in the opposite direction. This asymmetry in the movement is necessary, because otherwise counting errors especially in the start-up phase in each movement might cancel each other out. This is only true however, since counting errors caused by spikes are never negative.

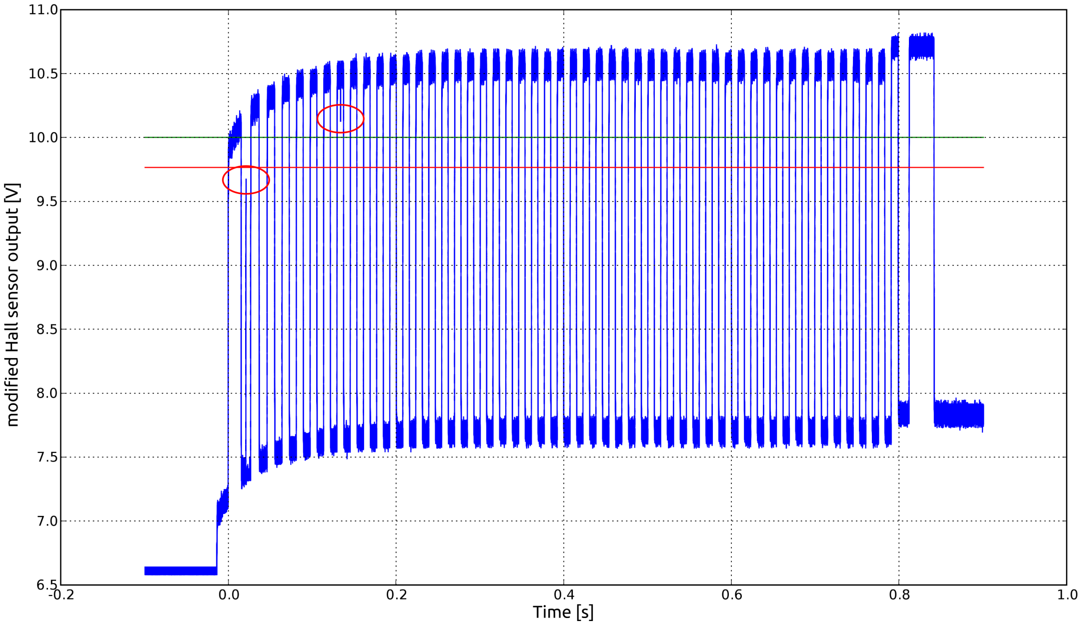


Figure 6.6: A sample of the output of a hallsensor. When selecting thresholds, the different levels at the beginning must be taken into account. The algorithm must also be able to handle spikes in the sensor output. Such spikes can be seen in the circled regions.

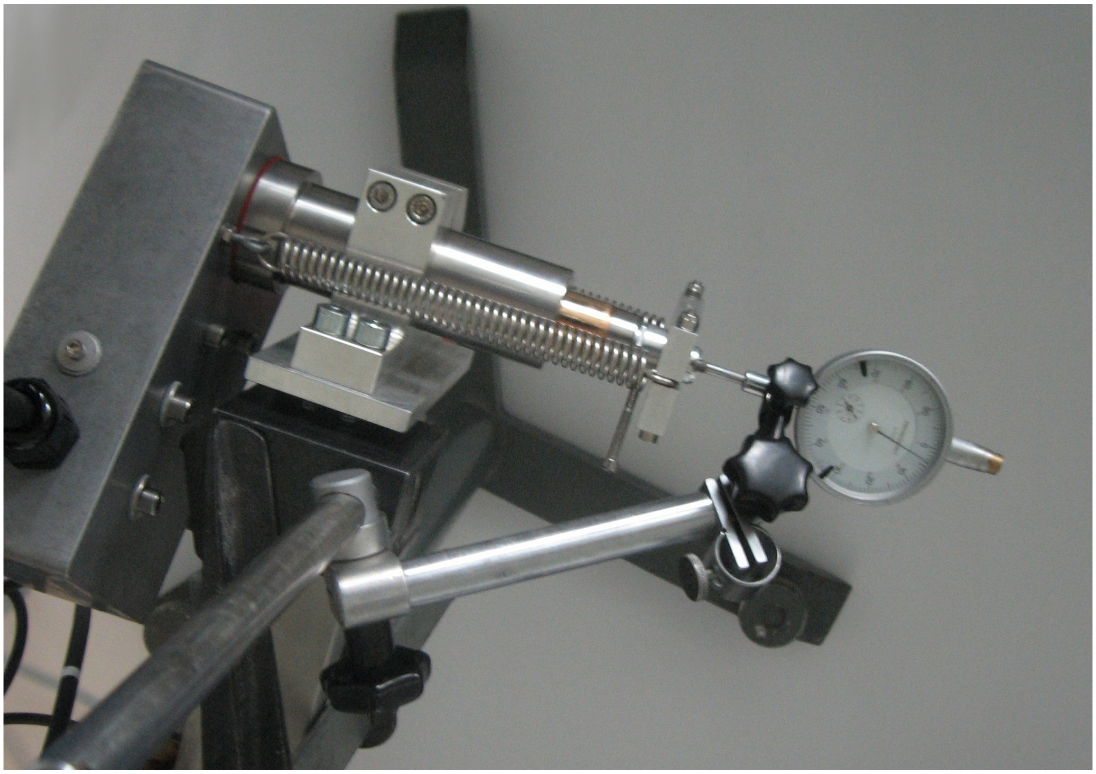


Figure 6.7: Test rig for long running hallcount detection tests. The difference between start- and endpositions was measured. One mark on the dial gauge is 0.01 mm, and the mechanical play in the whole setup is also about ±0.01 mm.

Several parameters were used in the tests. Initially, 100 counts up and 100 counts down were repeated 200 times. An offset was not measurable. The same was done with 200 counts and 500 counts repeated 500 and 1000 times. In one measurement a difference of about 0.01 mm occured.

After these initial symmetric tests, the asymmetric tests were started with 100 counts up and 2*50 counts down repeated 500 times. Two of 6 measurements showed again an offset of about 0.01 mm in the end. Since the precision of the mechanical setup is also about ± 0.01 mm, the result is that no counting error could be measured beyond the precision of the test rig.

In the end a single asymmetric long-running test with ~ 4400 cycles was conducted that again showed no measurable offset. The number was chosen to demonstrate that with 2 cycles per observation day, 22 observation days per month, and for 10 rain-free month per year, the system can run without a counting error for 10 years.[1]

6.4.4 Motor Board Controller

The motor board controller processes are used to control the selection of the motor board, the motor within that board via the multiplexers and demultiplexers on the motor boards. The processes are also responsible for setting the user logic into several distinct states: *stopped*, *cleared*, *running*, *debug*, *temperature sensor 1 - 4*, and *error*.

The MOSFET on the panel bus is switched on during the first *running* or *debug* state, which enables the power to the motor boards.

During *stopped*, *running* and *debug* state, the hallsensor is analyzed to not miss movements of the motor. But only during *running* state the number of detected hallcounts will be used as a switch off criterion. During *stopped* state, this switch off should not be necessary, since the motors are already stopped, but the hallsensors are still monitored since the motor continues to run for up to four hallcounts after the break signal (although a typical value is three hallcounts when run in the direction of the spring force, and two hallcounts when run in the opposite direction). During *debug* state, no automatic switch off is applied. This state is intended for debugging only.

During the *running* state two other criteria are used for stopping the motor. One is the signal from the current sensor, the other is the time between two hallsensor edges. Both stopping criteria can not be applied in the beginning of the motor movement, since the current is much higher for short time, and the motor movement is slower at the beginning which leads to a larger time between two hallcounts. The hardware deals differently with both situations: the power threshold is simply only checked after a (configurable) number of detected hallcounts. Before that, the power threshold is ignored. The current default number of hallcounts until the power threshold sets in is 8. The second condition, the time between two hallsensor edges, is the only stopping criterion at the beginning of the motor movement. To accommodate for the slower movement at the beginning, two

[1] This was however no requirement for MACS II: one hallcount results in about 0.01 mrad light offset. The pixel size in H.E.S.S. Phase 2 is 1.2 mrad and therefore about 100 hallcounts wide.

different time thresholds are implemented. The switch between the time thresholds is done at the same time that the power threshold is set up. The initial threshold is set to be 300 ms, the later threshold during normal movement is set to a much lower value of 11 ms.

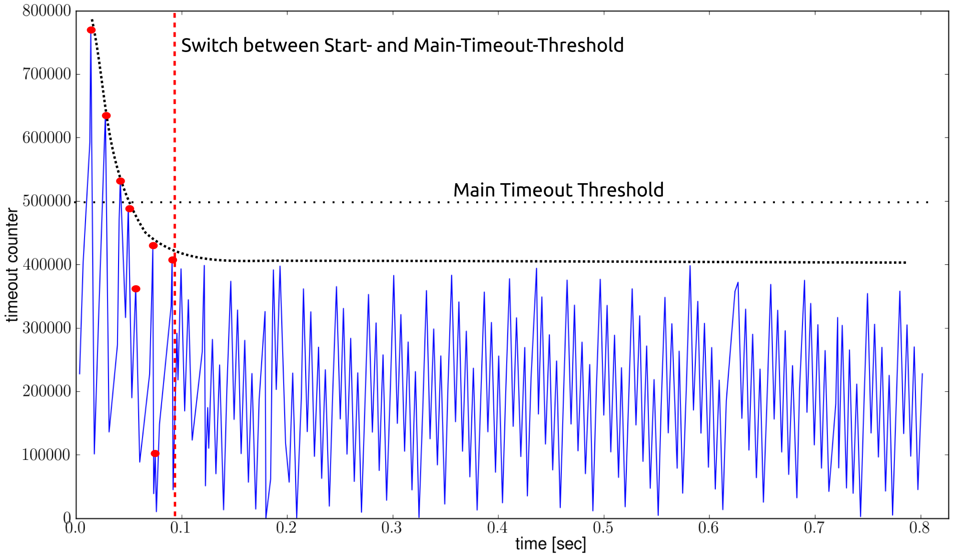


Figure 6.8: Software recording of the timeout counter between two consecutive hallcounts. Since this is a software recording, there is an unknown offset between the production and the recording of the value due to bus- and network-traffic and CPU scheduling. A peak in the line means that a hallcount was detected between two recordings, it does not mean that this is the exact value of the timeout counter at the hallcount detection, which is almost always higher. The red dots show the first eight hallcounts after which the switch between the start timeout threshold and the main timeout threshold takes place.

6.5 *Hardware Testing*

All hardware components have been tested prior to shipment to the H.E.S.S. site in Namibia. Since MACS II consists of many components, this was a major task for the IAAT and was mainly carried out by staff of the institute as well as by student assistants. The 2000 actuators were all driven over the complete intended range (4 cm) and the current as well as the hallsensor signal was monitored. This was necessary, since it ensures that the spindles in the actuators are not deformed, a problem that was discovered in some of the actuators. If a spindle is deformed, this can lead to an increased current, and ultimately to the destruction of the thread.

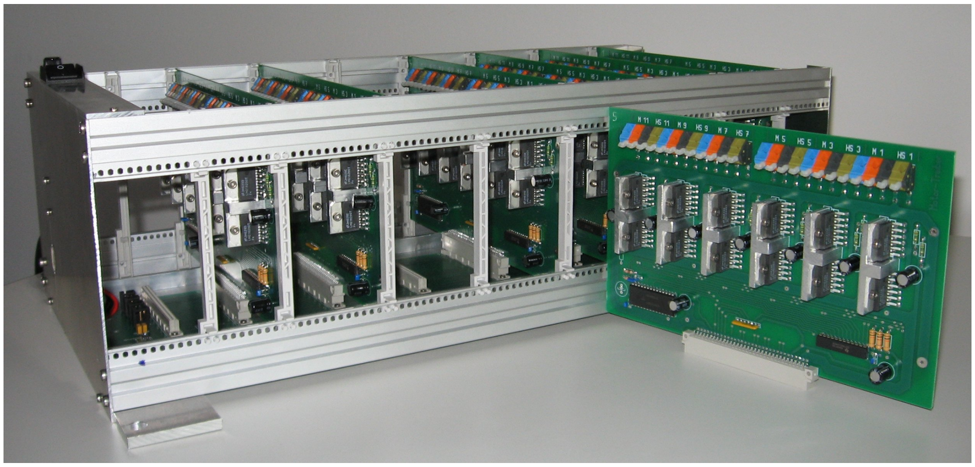


Figure 6.9: The interior of a panel controller box. The media converter cannot be seen from this angle.

All 2000 motors have also been tested individually, to ensure that the hallsensors are working correctly. Possible problems could have been missing permanent magnets or too high current consumptions. Fortunately, no motors showed these problems.

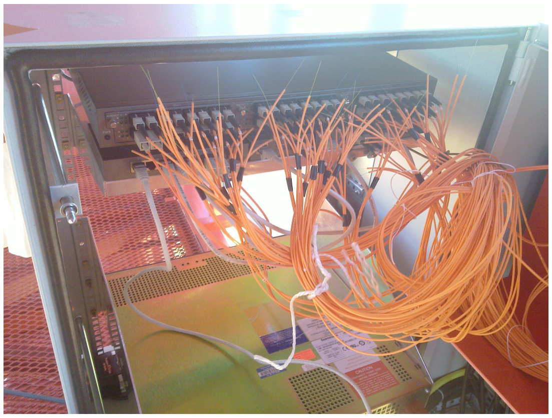


Figure 6.10: The media converter in the central box and the optical fibres coming from the panel controller boxes.

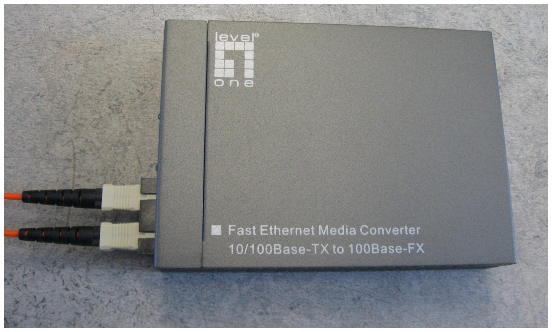


Figure 6.11: The media converter that is used in each panel controller box.

The more complex parts of the mirror alignment system are the panel controller boxes (see Figure 6.9). Since they include several different components, there are many things that potentially couldn't work, from ICs like multiplexers, over the Suzaku board and its configuration, to the media converter boxes (see Figure 6.10).

Before the shipment to Namibia, all panel controller boxes have been tested. To do this, we connected a test motor on every position on the motor boards where a motor will be actually be plugged in when installed on the telescope. The movement of the motor was recorded with a PC. The communication involved the TCP/IP connection via the media converter and the Ethernet cable to the Suzaku board.

The configuration of the Suzaku board was tested as well during this process. This was necessary, since each uClinux installation has its own fixed IP address, which is configured during the image creation process. Since each installation has had a complete re-compilation, they had to be tested separately. Additionally, this test also included the panel controller bus board and the motor boards.

A few hardware defects were detected and corrected during the tests, mostly due to malfunctioning ICs. Also, the thresholds of the timeout criterion needed to be adapted, since a few times, the motors required a bit more time to settle into the main, constant part of the movement.

6.6 Actuator Mounting and Cabling Campaign

In November and December 2011, a group from the IAAT installed the actuators, control boxes and cables on CT5 in Namibia. The mounting campaign prepared the telescope for the installation of up to 877 mirrors.

The campaign was a success, since all components could be mounted within the given time frame. As a final measure after the installation, every motor was driven via the alignment system. This tested the mechanical installation and cabling, but also the software implementation and communication as a complete system. The system worked perfectly in these tests.

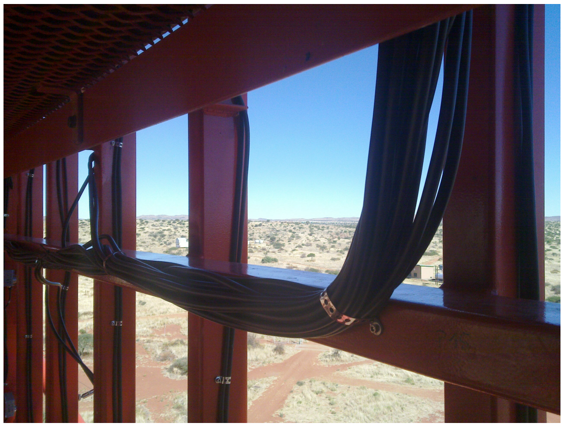


Figure 6.12: Power cables in CT5. Metal stripes are screwed to the steel structure to fixate the cables.

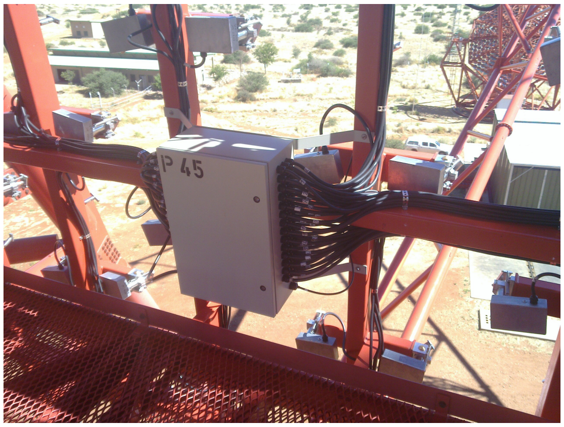


Figure 6.13: A panel controller box in CT5. Power and signal cables connect up to 84 motors with each box. One optical cable leads to the central box.

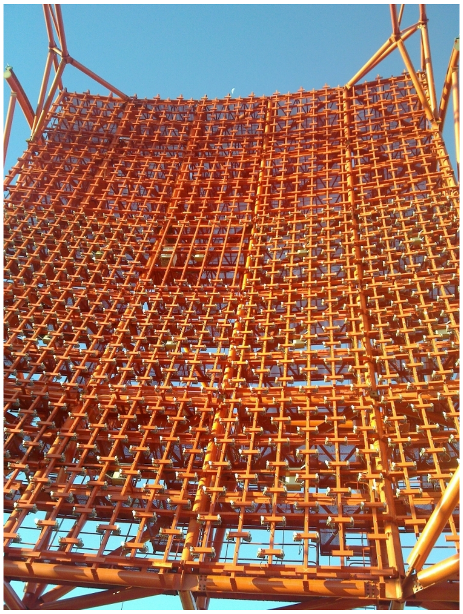


Figure 6.14: The mirror support structure of CT5 during the mounting campaign. The lower 15 panels were already finished, while the upper 10 panels still had to be worked on.

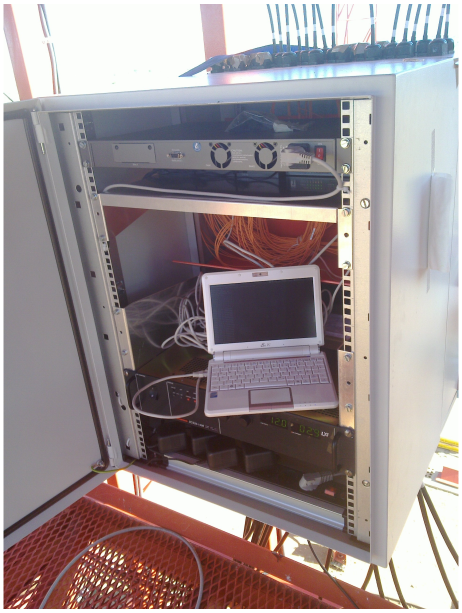


Figure 6.15: The central box during the finalizing tests of the mounting campaign. A netbook was used for these tests in the telescope.

Chapter 7

Software Development for MACS II

The software written for MACS II can be divided into two parts: the software running on the Suzaku boards and the central control software on the computer outside the telescope.[1] The part on the Suzaku board must fulfill special requirements, because the CPU used in this System-on-a-chip and the specific operating system are quite different from any off-the-shelf PC style system. The main difference comes from the fact that most modern CPUs include a component that is called a Memory Management Unit (MMU) while the Microblaze CPU on the Suzaku board does not have this component. Development of software for MMUless embedded systems is much more complicated then for normal PC systems (see Chapter 7.1 below). The other big challenge arises from the limited resources: a low CPU frequency of ~ 50 MHz and only 4 MB for the hardware configuration, the uClinux kernel and all custom software. For these and other reasons, the software on the Suzaku board had to be kept very simple and most logic was moved either to the central control software that controls the alignment process or to the hardware if real-time speed was required.
The software components will be described in the following paragraphs.

7.1 The uClinux Kernel and the Suzaku Server

uClinux (pronounced 'you-see-linux') is the operating system on the Suzaku board. The 'uC' in uClinux stands for μ (as in micro) and 'C' as in 'controller'. Since μ is not part of the ASCII letters, u is used instead throughout the literature and in the programming community. uClinux was originally developed as a port of the 2.0 branch of the Linux kernel for micro-controllers without MMU. The development was started in 1998 by D. Jeff Dionne and Michael Durrant from Arcturus Networks. An introduction to development with uClinux can be found in Brinker et al. (2007).
All modern CPU architectures for PCs and servers contain a MMU as part of the CPU chip. The MMU is a complex structure that is often of the same size and complexity as the actual CPU. This is the reason why in embedded CPU architectures like the Microblaze, the MMU is omitted. The responsibility of a MMU is to translate the memory addresses of processes on the CPU into physical addresses, either in memory or on the disk (see e.g. Opdenacker 2007). All memory addresses in compiled software for larger CPUs are

[1] As mentioned earlier, the complete project – the suzaku-s files including the newly added components as well as the software components – are available to IAAT members via a git repository at: `git@aithp1.ait.physik.uni-tuebingen.de:thesis/MACSII.git`.

therefore virtual, that means that they don't point to the real location in memory, but to a virtual location that is translated by the MMU into the real physical address. This can be seen in Figures 7.1 and 7.2. During the process of translation from virtual to physical addresses, the MMU also makes sure that processes can not access the memory of other processes.

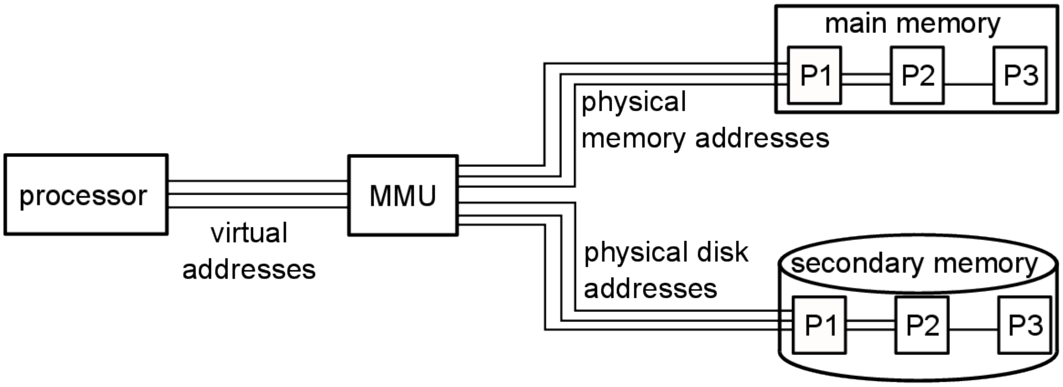


Figure 7.1: Memory access by processes on a system with a MMU. The processes use virtual addresses which get translated by the MMU into physical addresses in memory or on disks. The physical addresses are unreachable by the processes, and virtual addresses that would result in memory regions of other processes are forbidden by the MMU. Violations of this rule result in segmentation faults, which protect the memory of other processes.

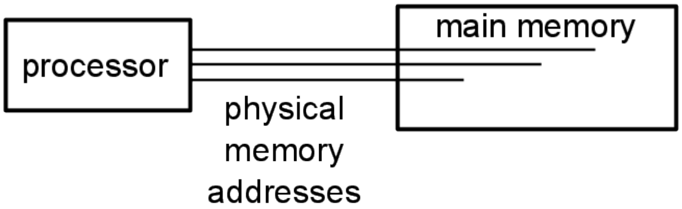


Figure 7.2: On a system without a MMU, the processes need to access the memory by their physical addresses. Nothing protects the memory from being accessed by other processes which are not supposed to access the memory. Because of this, on a system without MMU software running in user space can access even hardware without the intermediate step of kernel drivers.

This sandboxing of process memory has brought a lot of stability and security to modern operating systems and software, because a single bug in one program can not corrupt the memory of another program. There are tools like Valgrind which can help the programmer spotting places in the software, where these forbidden memory accesses will happen.

On CPUs without a MMU on the other hand, a memory address is an absolute address to the physical location of the memory cells. So each program can access the memory of every other program if it either knows the absolute memory address or by accidentally using an address that is already used by another program. This has severe consequences for the design of the software. The most unpleasant one for the programmer is a new kind of error that can be introduced: when one program writes in the address space of another program, the second program might continue to run undisturbed for an unknown period of time, until it tries to read that piece of memory. If this happens, the second program can show any number of behaviours from totally crashing to behaviour changes, that are indistinguishable from the intended behaviour. For example, it could have stored a value of 2 in a variable. After the access from the other program, the variable could now contain the value 5. Changes of this kind are extremely hard to detect. And bugs of this kind a therefore hard to debug, because they might happen at times unrelated to the execution time of the buggy code and within a program unrelated to the program containing the bug. Software for MMUless systems can for these reasons not be fully analyzed and debugged by memory analyzers like Valgrind.

In modern operating systems, there is a separation of kernel space and user space. Software running in the user space can not access the hardware directly. The possibility to do so is reserved for software running in kernel space. Developers that need access to hardware components have to write kernel drivers and separate software that can then use the kernel drivers.

The simpler development model of software for a MMUless CPUs is that every process can access the kernel space and therefore every process can access hardware if the physical hardware addresses are known.

On the Suzaku board, there is a defined memory range (from 0x81000000 to 0x81000058) that is used to synchronize the registers between the custom hardware and the software. The other variables that are used in the software must be assigned fixed memory addresses during the compilation step. This requires a custom compiler that knows which parts of the physical memory are free to use on the specific Suzaku board. This also means that the software can not create new variables at run-time with the *malloc* command family.

The MACS II software on the Suzaku board, short ‘Suzaku server’, is responsible for providing TCP/IP access to the MACS II hardware. uClinux, as deployed on the Suzaku board, provides a complete TCP/IP stack.

The communication part of the Suzaku server is implemented as a single threaded server. The reason (again) is the missing MMU, which makes *fork* unavailable on these systems. Therefore, only one connection at a time between the server on the Suzaku board and other software is possible.

Although not the most modern language, C is one of the languages which provide compilers for nearly every hardware. The Suzaku server is therefore implemented in C and compiled with the Microblaze GNU Compiler Collection (mb-gcc).

7.2 The Communication Between the MACS II Components

A major reason for the chosen setup of the whole mirror alignment system and the selection of its components was the early decision that the communication between the central control software and mirror control components on the telescope should be done via standard Ethernet protocols.

On the already working telescope of H.E.S.S. Phase 1, twelve strings of (power) cables are going from the electronic hut at the bottom of each telescope into the telescope dish. This kind of setup would not work for the much larger number of mirrors (see Chapter 3.4). These require longer cables which address and power all the actuators of the many more mirrors. At the same time, while the H.E.S.S. Phase 1 telescopes have external lightning protection masts, the H.E.S.S. Phase 2 telescope will not.

Both arguments, the higher mirror number and the missing lightning protection, pointed at a different overall setup for the mirror alignment system, namely one that minimized the number of cables that have to go into the telescope. Such a setup means that there must be a communication center on the telescope which distributes the power and monitors the motors. Since the H.E.S.S. Phase 2 telescope dish is made up of 25 identical panels, this separation was also transferred into the mirror alignment system and 25 identical communication centers where build. Since each center needs to communicate with the central control software, a fast and reliable communication means was chosen: Ethernet with the TCP/IP protocol.

The use of such a standard communication protocol has the effect, that it can be accessed from nearly every programming language. This would make transitions of either component, the Suzaku server or the central control software, much easier than custom protocols would do. Similar considerations were also applied by other developers of H.E.S.S. Phase 2 components, e.g. the CCD system (see Chapter 7.3.4).

The communication between the two software components of MACS II is based on TCP/IP with a custom protocol on top.[1] All commands of the custom protocol are ASCII based and newline terminated. The limitation to ASCII characters instead of binary values like integers and floats, increases the message size slightly but this is a minor drawback, since most messages require only a few small numbers and the messages in general are small compared to the TCP/IP overhead. The major gain of this restriction is the reduction in error potential that single-byte entities have over multi-byte entities, since (as explained earlier) the MACS II setup includes 3 different bit- and byteorder systems.

The protocol of the communication between the Suzaku server and the central control software is composed of requests and answers. Each request from the central control software is followed by an answer from the server on the Suzaku board. The protocol can be described in modified BNF[2] in the following way.

The general communication structure is shown in Listing 7.1 below.

[1] 'A custom protocol on top of TCP/IP' is sometimes also called 'API' (Application Programming Interface).

[2] Backus-Naur-Form; Modifications are: A star (*) means zero or more repetitions of the preceding item; likewise, a plus (+) means one or more repetitions. Both operators bind as tightly as possible. Parentheses are used for grouping. Two literal characters separated by three dots mean a choice of any single character in the given

Listing 7.1: General protocol structure

```
<request> ::= (<stop> | <run> | <get> | <set>) <EOL>
<answer>  ::= (<ok> | <error> ) <EOL>
<ok>      ::= ’ok’ (<stop_answer> | <run_answer>  |
                    <get_answer>  | <set_answer>) |
                    <recording>* <end_recording>
<error>   ::= ’err’ <error_description>
```

There are three different ways to control the motor movement. The most common one is to start the motor and check periodically if the motor movement has stopped. During the development phase, or if errors need to be debugged on the telescope, two other ways are implemented. Both stream a recording of the ADC output to the central control software. The first mode allows the hardware to still apply the stopping mechanisms, the second one causes an overwrite of the ADC control mechanisms, which completely removes the hardware control over the automatic stopping mechanisms. The second command should only be used in a laboratory environment.

The other commands in the protocol provide an interface to the software readable/writable parameters. An overview is given in Listing 7.2.

Listing 7.2: Motor movement and parameter requests

```
<run>    ::= <simple> | <record> | <debug>
<simple> ::= ’1’ <id>  <board> <motor> <dir> <counts>
<record> ::= ’4’ <id>  <board> <motor> <dir> <counts>
<debug>  ::= ’5’ <adc> <board> <motor> <dir> <counts>
<adc>    ::= ’0’...’11’

<stop>   ::= ’0’ (<keep> | <clear>)
<keep>   ::= ’1’
<clear>  ::= ’0’

<get>    ::= ’3’ <parameter>
<set>    ::= ’2’ <parameter> <value>
```

Although not visible in the listing above, there is a multi-line answer in the *recording run* and *debug run* case. In these cases, the software on the Suzaku board sends a stream of register contents, and closes the stream with an ‘ok done <EOL>’, so the protocol remains newline based. To shorten the description of the protocol, the definition of parameters were omitted. All parameters are ASCII representations of integers, except for status, which is given as an ASCII representation of a 32 bit binary value.

(inclusive) range of ASCII characters. Numbers can be defined the same way, although multi-character numbers can be given with a start and end number.

7.3 The Central Control Software

The central control software (also called *macsII client*, since it acts as a client to the server on the Suzaku board) is responsible for:

- providing a user interface to control the alignment of the mirrors
- implementing the alignment algorithm
- extracting information from and storing information in the mirror database
- communication with the Suzaku server
- accessing the Lid-CCD camera server

7.3.1 The Development Tools

While the tools that could be used in the FPGA development (Xilinxs EDK, VHDL) where already chosen by the vendors of the Suzaku board, there was much more freedom with the choice of the software development tools. The most important one is the language, and like other developers of recent H.E.S.S. software, I have chosen Python as the language for the Central Control Software (specifically Python 2.x, since it depends on numpy, scipy, pyfits and matplotlib which are not yet ported to Python 3.x). The Python code is formatted according to PEP8 (van Rossum and Warsaw 2001), unless readability of the code would otherwise suffer.
During the development, code changes were controlled with the *Bazaar*[1] distributed version control system. To keep track of the bugs in the code, the distributed bug tracker *Bugs-Everywhere*[2] was used. It stores the bug data local in the version repository instead of requiring a central server as most other bug trackers do.
The documentation was created with the help of *Sphinx*[3], a tool that re-formats documentation written in reStructuredText into html and pdf (via latex).
During the development process, the code was structured into independent units which can then be tested separatly and in an automatic way. This automatic testing was done with the help of the python module *unittest* and the *nosetest*[4] framework. The coverage report of code statements that can be tested by running the test suite can be seen in Table 7.1.

Apart from ensuring that the software works as expected, this automatic test suite can also be a great help when applying changes to the code base in the future. Since the scientists involved in H.E.S.S. Phase 2 components like the mirror alignment are often PhD students and therefore likely to take turns regularly, any future developer can make changes to the code base and be sure that his changes have not introduced incompatibilities to other parts of the code if the automatic tests still work.

[1] Bazaar: `http://bazaar.canonical.com/en/`

[2] Bugs-Everywhere: `http://bugseverywhere.org/`

[3] Sphinx: `http://sphinx.pocoo.org/index.html`

[4] Nose: `http://readthedocs.org/docs/nose/en/latest/`

Table 7.1: Automatic test coverage of macsII (State: 2011-09-03). Modules marked with a star (★) contain much code that needs external communication partners like the CCD server or the Suzaku servers and have therefore been tested mainly manually.

Python Module	Statements covered by automatic tests
macsII.alignment_helpers	96%
macsII.macsIIconfig	92%
macsII.alignment	81%
macsII.macsII_cli	81%
macsII.ccd	78%★
macsII.models	64%
macsII.panelcontroller	27%★
macsII.macsII_dummy_ccd_server	23%★

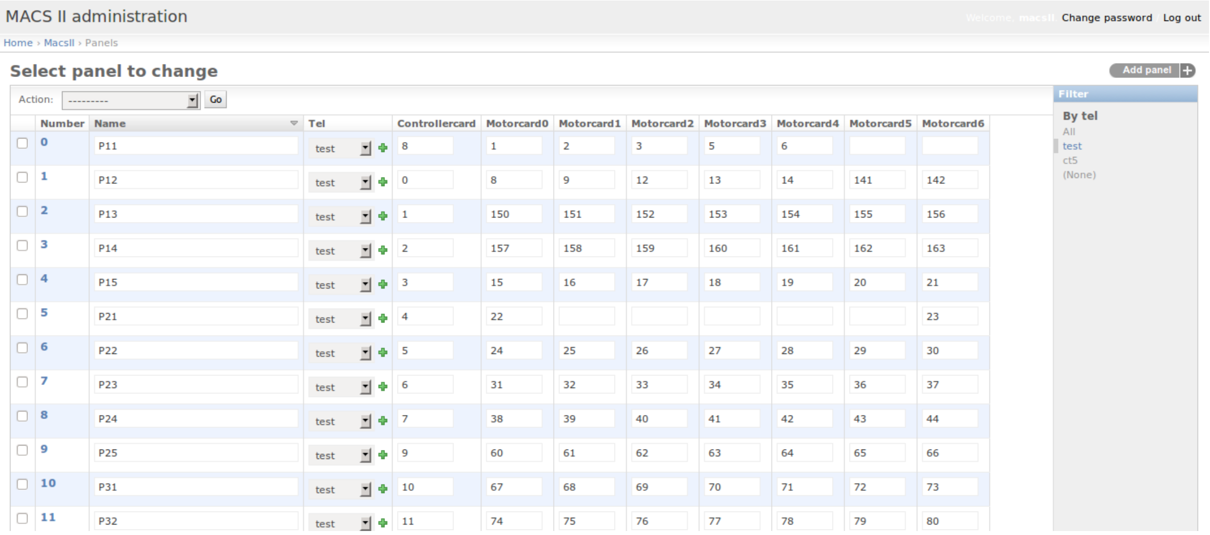
7.3.2 The Database

During the preparation and operation of MACS II, information about the mirror positions and movements need to be permanently stored. Possible solutions for data storage are simple files or a database. For MACS II, the latter approach was taken, mainly for speed reasons, but also to avoid reimplementing typical database operations when they are already available. To also avoid a lot of handwritten SQL code, a so called Object-Relational-Mapper (ORM) system was used. From the many options (SQLalchemy, Storm, ...), I have chosen to use the Django database abstraction layer[1], which allows to define objects with properties in Python, which are then mapped by Django to a number of different possible databases. For MACS II, the chosen database is SQlite[2], but this decision is completely separated from the code, so that a different database like MySQL or PostgreSQL could be used without the need to change the code (the data however would have to be transferred).

The MACS II database contains tables for the telescopes (because there is at least one additional *test* telescope and a *test_reference* telescope, both are used for automatic testing and for letting new users learn about the alignment), the panels, and the mirrors. The ‘panel’ tables contain all the information that are needed to communicate with the Suzaku servers on the telescope (like the port, IP addresses, but also housekeeping information about the hardware e.g. identification numbers in the panel boxes). The ‘mirrors’ table is used to store information about the individual states (referenced, broken, aligned, ...) and the transformation matrix of the mirror. It also contains information about the positions, alignment positions, and possible ranges of the actuators.

[1] `https://www.djangoproject.com/`

[2] `http://www.sqlite.org/`



MACS II administration

Select panel to change

	Number	Name	Tel	Controllercard	Motorcard0	Motorcard1	Motorcard2	Motorcard3	Motorcard4	Motorcard5	Motorcard6
□	0	P11	test	8	1	2	3	5	6		
□	1	P12	test	0	8	9	12	13	14	141	142
□	2	P13	test	1	150	151	152	153	154	155	156
□	3	P14	test	2	157	158	159	160	161	162	163
□	4	P15	test	3	15	16	17	18	19	20	21
□	5	P21	test	4	22						23
□	6	P22	test	5	24	25	26	27	28	29	30
□	7	P23	test	6	31	32	33	34	35	36	37
□	8	P24	test	7	38	39	40	41	42	43	44
□	9	P25	test	9	60	61	62	63	64	65	66
□	10	P31	test	10	67	68	69	70	71	72	73
□	11	P32	test	11	74	75	76	77	78	79	80

Figure 7.3: A graphical interface to the mirror and panel database is also provided by the MACS II web server.

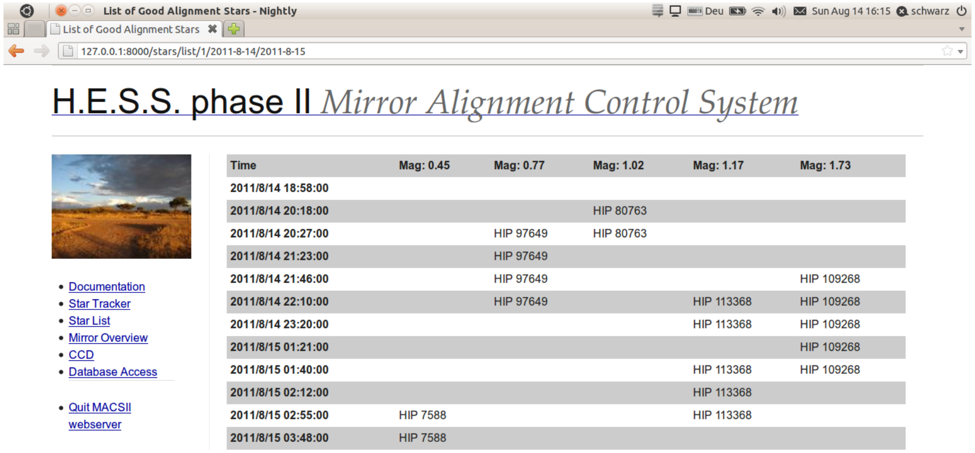
7.3.3 The User Interface

The initial plan for MACS II was to provide a Graphical User Interface (GUI) that would allow new users to start easily with the alignment of mirrors. During the alignment campaigns of H.E.S.S. Phase 1 however, I have learned that a command-line interface for the actual alignment might still be the better solution.

Therefore MACS II includes both, a graphical user interface as well as a command line interface.

Situations in which a command-line interface to the alignment software is needed include standing in front of the telescope, with a netbook or smartphone in the hand and trying to move single mirrors that could not be aligned automatically at night from the control room. In such situations, the network connection can be very bad. Low-bandwidth interfaces that require only reduced mouse/touchscreen interaction are preferable, then. Another aspect is the scriptability, which makes a command-line interface for the actual alignment with MACS II the better choice. Therefore all the major tasks for database access and alignment were implemented as command-line options of the *macsII* software.

The graphical interface offers a tool with multiple functions for the user sitting in the control room having a good internet connection.
The user gets access to the mirror alignment control system by starting the MACS II webserver and visiting the welcome page. The first thing needed for a successful alignment is a list of suitable stars where the telescope can be pointed at. Planning an alignment therefore requires information about positions of a selected number of bright and isolated stars in a defined zenith angle range.



List of Good Alignment Stars - Nightly

127.0.0.1:8000/stars/list/1/2011-8-14/2011-8-15

H.E.S.S. phase II *Mirror Alignment Control System*

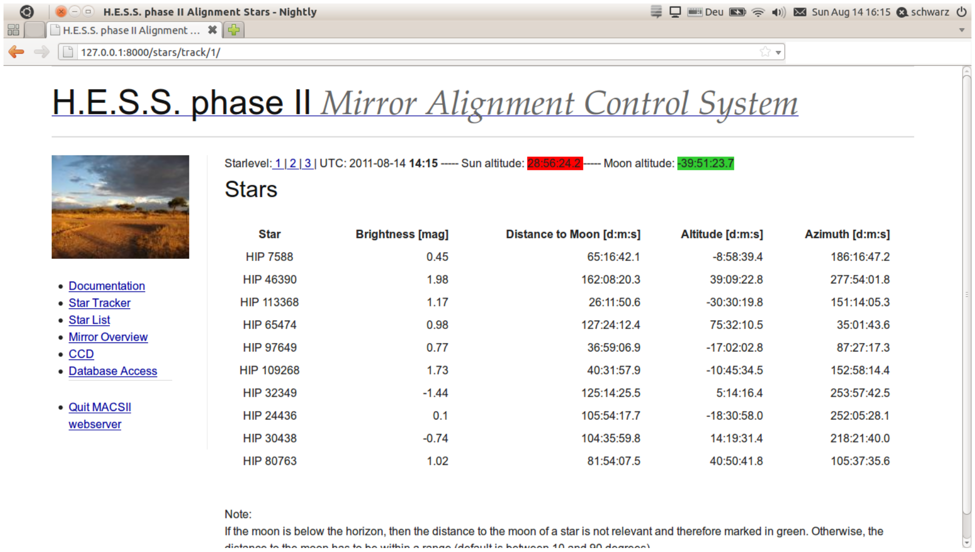
- Documentation
- Star Tracker
- Star List
- Mirror Overview
- CCD
- Database Access

- Quit MACSII webserver

Time	Mag: 0.45	Mag: 0.77	Mag: 1.02	Mag: 1.17	Mag: 1.73
2011/8/14 18:58:00					
2011/8/14 20:18:00			HIP 80763		
2011/8/14 20:27:00		HIP 97649	HIP 80763		
2011/8/14 21:23:00		HIP 97649			
2011/8/14 21:46:00		HIP 97649			HIP 109268
2011/8/14 22:10:00		HIP 97649		HIP 113368	HIP 109268
2011/8/14 23:20:00				HIP 113368	HIP 109268
2011/8/15 01:21:00					HIP 109268
2011/8/15 01:40:00				HIP 113368	HIP 109268
2011/8/15 02:12:00				HIP 113368	
2011/8/15 02:55:00	HIP 7588			HIP 113368	
2011/8/15 03:48:00	HIP 7588				

Figure 7.4: Step 1 in planning an alignment night: finding the brightest stars with the star list tool.

The list of such stars was compiled by Cornils (2006) for the H.E.S.S. Phase 1 macs. The user can access this list on the welcome page of the alignment control system. Likewise, the interface provides information which is needed during the alignment, namely the position of the stars and their distance to the moon as well as the current position of the sun.



H.E.S.S. phase II Alignment Stars - Nightly

127.0.0.1:8000/stars/track/1/

H.E.S.S. phase II *Mirror Alignment Control System*

- Documentation
- Star Tracker
- Star List
- Mirror Overview
- CCD
- Database Access

- Quit MACSII webserver

Starlevel: 1 | 2 | 3 | UTC: 2011-08-14 **14:15** ----- Sun altitude: 28:56:24.2 ----- Moon altitude: -39:51:23.7

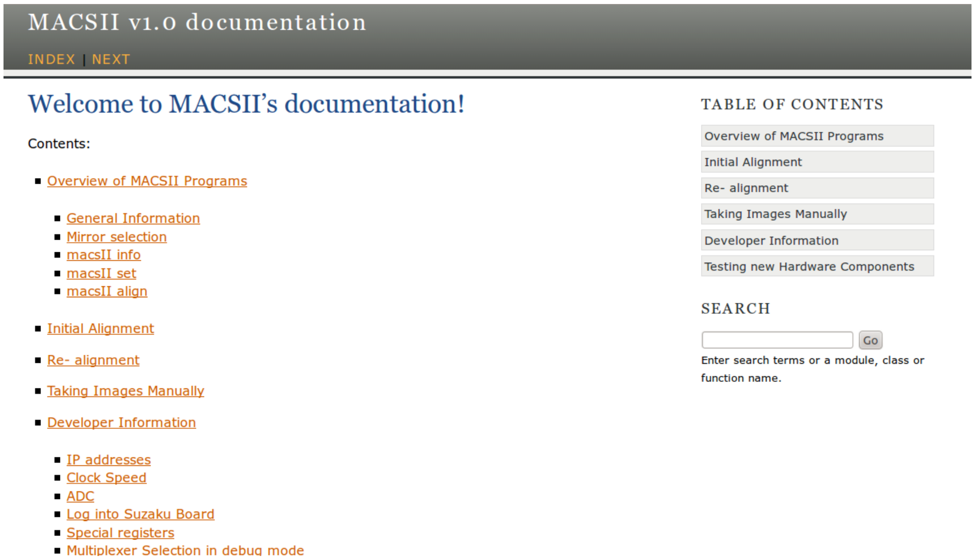
Stars

Star	Brightness [mag]	Distance to Moon [d:m:s]	Altitude [d:m:s]	Azimuth [d:m:s]
HIP 7588	0.45	65:16:42.1	-8:58:39.4	186:16:47.2
HIP 46390	1.98	162:08:20.3	39:09:22.8	277:54:01.8
HIP 113368	1.17	26:11:50.6	-30:30:19.8	151:14:05.3
HIP 65474	0.98	127:24:12.4	75:32:10.5	35:01:43.6
HIP 97649	0.77	36:59:06.9	-17:02:02.8	87:27:17.3
HIP 109268	1.73	40:31:57.9	-10:45:34.5	152:58:14.4
HIP 32349	-1.44	125:14:25.5	5:14:16.4	253:57:42.5
HIP 24436	0.1	105:54:17.7	-18:30:58.0	252:05:28.1
HIP 30438	-0.74	104:35:59.8	14:19:31.4	218:21:40.0
HIP 80763	1.02	81:54:07.5	40:50:41.8	105:37:35.6

Note:
If the moon is below the horizon, then the distance to the moon of a star is not relevant and therefore marked in green. Otherwise, the distance to the moon has to be within a range (default is between 10 and 90 degrees)

Figure 7.5: During the alignment: monitoring the position of the sun and the distance between stars and the moon with the star tracker tool.

Apart from the star list and the star tracker tools, the user can access all database information about the mirrors. He can for example inform himself about which mirrors still need alignment. He can also manually take pictures of the PMT camera lid.



MACSII v1.0 documentation

INDEX | NEXT

Welcome to MACSII's documentation!

Contents:

- Overview of MACSII Programs
  - General Information
  - Mirror selection
  - macsII info
  - macsII set
  - macsII align
- Initial Alignment
- Re- alignment
- Taking Images Manually
- Developer Information
  - IP addresses
  - Clock Speed
  - ADC
  - Log into Suzaku Board
  - Special registers
  - Multiplexer Selection in debug mode

TABLE OF CONTENTS

- Overview of MACSII Programs
- Initial Alignment
- Re- alignment
- Taking Images Manually
- Developer Information
- Testing new Hardware Components

SEARCH

Enter search terms or a module, class or function name.

Figure 7.6: All documentation of the MACS II commands and the documentation of the hardware components is available as websites and as a pdf document.

The documentation link leads to the complete documentation of MACS II, that means the commands and parameters of the software, but also to a lot of information about the hardware components (see Figure 7.6).

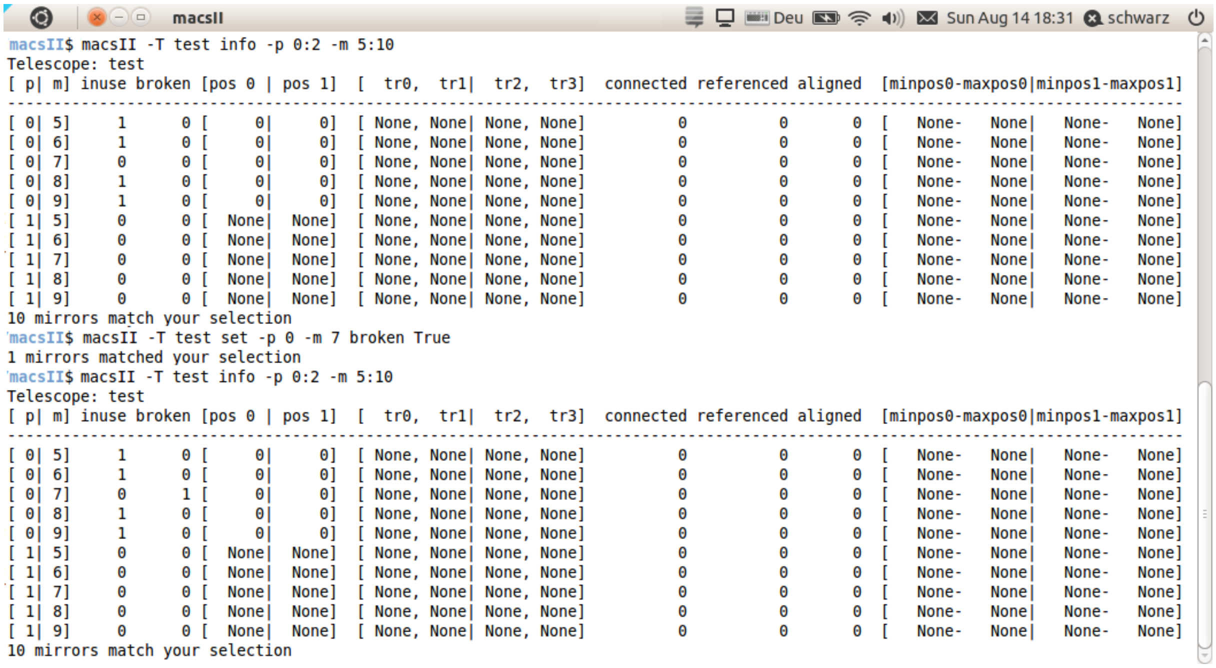
The actual alignment is not done with the described graphical interface (even though information needed for the alingment can be found there) but with the command-line interface.

In the command-line the user can start the automatic alignment process. This process follows an alignment algorithm which is described in the following chapters in greater detail. Via the command-line the user can either align the mirrors individually or align a whole selection of mirrors.

The central command entered into the command line is: *macsII -T ct5 align*. This command starts a complete initial alignment of the whole telescope.

The base command *macsII* accepts sub-commands for database information retrieval as well as sub-commands for changing the values of several mirrors in the database by entering a single command which includes a range selection (e.g. the command: *-p 2 -m 20:31* would select the mirrors 20 to 31 of panel 2 whereas the command: *-k connected -v False* would select additionally only mirrors that have not been connected so far.) Range selections like these can be used for all sub-commands including the alignment commands.

Besides entering alignment commands, the command-line interface offers a way to access the database information which is also accessable via the graphical interface and which is needed to get information about the position and status of each mirror (see Figure 7.7).



```
macsII$ macsII -T test info -p 0:2 -m 5:10
Telescope: test
[ p| m] inuse broken [pos 0 | pos 1]  [  tr0,  tr1|  tr2,  tr3]  connected referenced aligned  [minpos0-maxpos0|minpos1-maxpos1]
--------------------------------------------------------------------------------------------------------------------------------
[ 0| 5]     1      0 [     0|     0]  [ None, None| None, None]          0          0       0  [   None-   None|   None-   None]
[ 0| 6]     1      0 [     0|     0]  [ None, None| None, None]          0          0       0  [   None-   None|   None-   None]
[ 0| 7]     0      0 [     0|     0]  [ None, None| None, None]          0          0       0  [   None-   None|   None-   None]
[ 0| 8]     1      0 [     0|     0]  [ None, None| None, None]          0          0       0  [   None-   None|   None-   None]
[ 0| 9]     1      0 [     0|     0]  [ None, None| None, None]          0          0       0  [   None-   None|   None-   None]
[ 1| 5]     0      0 [  None|  None]  [ None, None| None, None]          0          0       0  [   None-   None|   None-   None]
[ 1| 6]     0      0 [  None|  None]  [ None, None| None, None]          0          0       0  [   None-   None|   None-   None]
[ 1| 7]     0      0 [  None|  None]  [ None, None| None, None]          0          0       0  [   None-   None|   None-   None]
[ 1| 8]     0      0 [  None|  None]  [ None, None| None, None]          0          0       0  [   None-   None|   None-   None]
[ 1| 9]     0      0 [  None|  None]  [ None, None| None, None]          0          0       0  [   None-   None|   None-   None]
10 mirrors match your selection
macsII$ macsII -T test set -p 0 -m 7 broken True
1 mirrors matched your selection
macsII$ macsII -T test info -p 0:2 -m 5:10
Telescope: test
[ p| m] inuse broken [pos 0 | pos 1]  [  tr0,  tr1|  tr2,  tr3]  connected referenced aligned  [minpos0-maxpos0|minpos1-maxpos1]
--------------------------------------------------------------------------------------------------------------------------------
[ 0| 5]     1      0 [     0|     0]  [ None, None| None, None]          0          0       0  [   None-   None|   None-   None]
[ 0| 6]     1      0 [     0|     0]  [ None, None| None, None]          0          0       0  [   None-   None|   None-   None]
[ 0| 7]     0      1 [     0|     0]  [ None, None| None, None]          0          0       0  [   None-   None|   None-   None]
[ 0| 8]     1      0 [     0|     0]  [ None, None| None, None]          0          0       0  [   None-   None|   None-   None]
[ 0| 9]     1      0 [     0|     0]  [ None, None| None, None]          0          0       0  [   None-   None|   None-   None]
[ 1| 5]     0      0 [  None|  None]  [ None, None| None, None]          0          0       0  [   None-   None|   None-   None]
[ 1| 6]     0      0 [  None|  None]  [ None, None| None, None]          0          0       0  [   None-   None|   None-   None]
[ 1| 7]     0      0 [  None|  None]  [ None, None| None, None]          0          0       0  [   None-   None|   None-   None]
[ 1| 8]     0      0 [  None|  None]  [ None, None| None, None]          0          0       0  [   None-   None|   None-   None]
[ 1| 9]     0      0 [  None|  None]  [ None, None| None, None]          0          0       0  [   None-   None|   None-   None]
10 mirrors match your selection
```

Figure 7.7: The database information which can be accessed via the command-line interface shows a list of the mirrors. It also shows for example whether the mirrors are in use or broken, and it gives information on the position of the actuators as well as information on the transformation matrices (see Chapter 7.4)

The command-line interface is fully self documented. The user can access all the documentation of the features and switches with the *macsII –help* command.
All in all, everything possible was done to create an easy to handle graphical as well as command-line interface for the H.E.S.S. staff that allows a (to the greatest possible extent) automatic alignment of the H.E.S.S. Phase 2 mirrors.

7.3.4 The CCD Interface

The Lid-CCD is used during the alignment, but it is not per se a component of MACS II, since it is mainly used for other tasks like precision pointing. For H.E.S.S. Phase 2, there will be a small computer in the camera housing that will provide a TCP/IP interface for all components that want to access the camera. The camera server was developed by D. Lennarz (MPI Heidelberg), J. Dreyling-Eschweiler, and A. Abramowski (both from the University of Hamburg). The server is written in C++ and provides an interface that is mainly ASCII based with newline characters as command separators. Unlike the MACS II protocol, however, it contains the transmission of binary data directly copied from memory. It additionally provides commands to query the variable size and byte order of the host system, to allow the clients to determine the exact binary protocol during run-time. Alternative implementations could have used a fixed protocol which would have required the server to convert from the native data types to the custom protocol types. But this was deemed unnecessary by the developers.

Since MACS II must communicate with the Lid-CCD, a Python client has been implemented for this server. The client was tested against a test Lid-CCD server setup running in Heidelberg. To be able to test the alignment further, a test server has also been implemented which complies with the protocol but returns simulated images with 'mirror spots' (simple 2D Gaussian with noise plus a background noise component) based on the current values of the actuators in the mirror database.

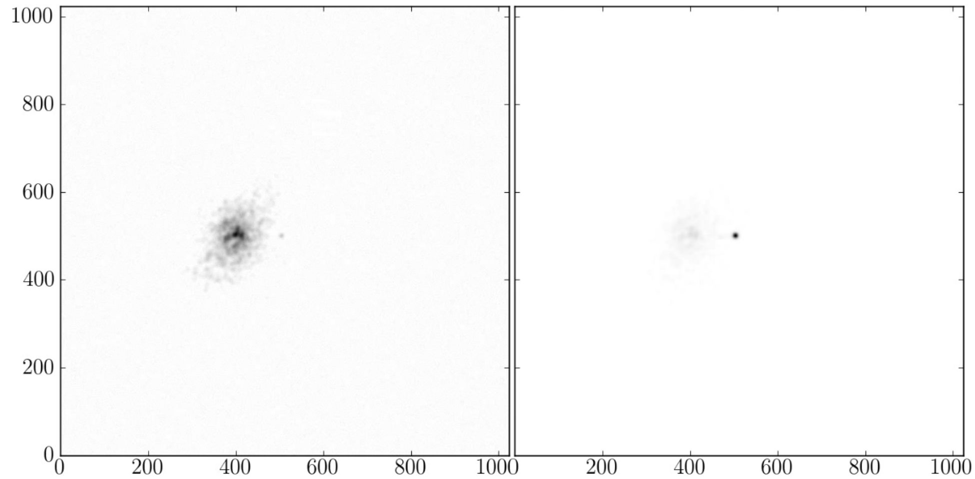


Figure 7.8: A test alignment with the custom test ccd server that creates spots on a simulated lid based on the values of the *test* telescope in the mirror database. The first random cloud of spots and a later stage of the alignment is shown. The alignment was carried out with the MACS II spot finding and mirror movement algorithm.

A simulated alignment of the test telescope in the database was carried out (see Figure 7.8) to test the spot finding and alignment algorithm, as well as the communication to the CCD server.

7.4 The Alignment Algorithm

The purpose of this part of the software is to detect the positions of the reflections of the individual mirrors in the CCD image. From these positions and from the movement of the actuators that lead to the displacement of the spots, the distance to the center of the PMT camera can be deduced, at least under the assumption of a linear correlation between the movement of the motors and the movement of the light spot.

7.4.1 A Mathematical Description

In mathematical terms, this approximation of the connection between motor movement and spot movement can be written in the following way (Cornils 2006):

$$\Delta x = T \Delta a \tag{7.1}$$

Here, $x \equiv (x_1, x_2)$ is the position of an individual mirror spot on the PMT camera lid and $a \equiv (a_1, a_2)$ are the positions of the actuators of that mirror. The transformation matrix

$$T \equiv \begin{pmatrix} \delta x_1/\delta a_1 & \delta x_1/\delta a_2 \\ \delta x_2/\delta a_1 & \delta x_2/\delta a_2 \end{pmatrix} \tag{7.2}$$

can be assumed to be even simpler if we take the geometry of the actuators into account. The tilting axes of a mirror are at angles of 120 degrees to each other, which means that the elements of the matrix are dependent:

$$\begin{pmatrix} \delta x_1/\delta a_1 \\ \delta x_2/\delta a_1 \end{pmatrix} \simeq R_{120} \begin{pmatrix} \delta x_1/\delta a_2 \\ \delta x_2/\delta a_2 \end{pmatrix} \tag{7.3}$$

Here, R_{120} is the rotation matrix for 120 degrees,

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, R(120) = \begin{bmatrix} -1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & -1/2 \end{bmatrix} \tag{7.4}$$

.

7.4.2 *The Basic Alignment Algorithm*

In H.E.S.S. Phase 1 the mathematical description shown above was translated directly into the alignment algorithm (Cornils 2006). The alignment was separated into a coarse and a fine part. The coarse alignment makes use of the assumption that the 120 degrees angle between the actuators is exact and leads to the simplified transformation matrix. In this case, only two spots are needed to determine the full matrix. Due to the approximation of the spot movement as linear function of the actuatormovement, this is only suitable for small movements. The fine alignment part used four spots around the center of the PMT camera lid. To get the spots close to the center, the coarse alignment had to be done first, so that the transformation matrix could be estimated.

The procedure (for the coarse alignment) was as follows:

1. Move all actuators such that no spots are in the FoV of the CCD camera
2. Move a single mirror such that the spot is likely to be in the FoV
3. Take a picture
4. Tilt the mirror a little
5. Take a second picture
6. Run the spot finding algorithm and calculate the transformation matrix
7. Move the mirror such that the spot is in the center
8. Repeat from step 2. for each mirror

Since only a single motor of a single mirror could move at the same time in H.E.S.S. Phase 1, this procedure could not be made any faster. Each mirror required two images, three movements and all that sequential. Per image roughly 45 sec were required plus some 10 seconds for the actuator movement.

The alignment of a complete H.E.S.S. Phase 1 telescope takes roughly 2 weeks. It must be noted however that the weather and the moon times play a critical role.

The separation into coarse and fine alignment is also present in MACS II. However, since in MACS II 25 mirrors can be moved at the same time (one in each panel), at least the coarse alignment time per mirror could be considerably reduced. The basic algorithm is still implemented in MACS II since it is very reliable. But another, faster algorithm is implemented as well (see Chapter 7.4.4).

7.4.3 The Spot Finding Algorithm

The spot finding algorithm for MACS II is implemented in Python (van Rossum et al. 2011) and is based on functions of the scipy and numpy libraries (Jones et al. 2011). Numpy is a library that provides fast N-dimensional array manipulation for Python with a nice and short syntax. Scipy is built on top of numpy and includes modules for e.g. statistics, Fourier transforms, signal and image processing.

To align the mirrors, it is necessary to analyse the image that the Lid-CCD takes of the spots that are cast by the mirror when they are pointed at a single star.

In reality it is hard to find a star with the following properties: bright, isolated, a zenith angle between $55°$ and $75°$, and a big enough distance to the moon ($10°$). Additionally, the region in the sky must be cloudless.

Cornils (2006) compiled an already mentioned list of possible alignment stars for H.E.S.S. Phase 1: a few suitable ones like Sirius (HIP 32349), and a number of stars, that are not so suitable, but might still be useful. In the later case, it is harder to automatically determine where the mirrors cast their spots, because a single mirror might make more than one spot (when the star is not isolated), or the camera lid might contain shadows of camera masts, cast by the moon (when the distance to the moon is not ideal).

Another challenge that the spot finding algorithm has to cope with is the variance of spot shapes of the individual mirrors. While an ideal mirror would cast a single, sharp spot, the majority of the mirrors for H.E.S.S. Phase 2 have elongated spots, sometimes even with several maxima. Four examples can be seen in Figure 7.9.

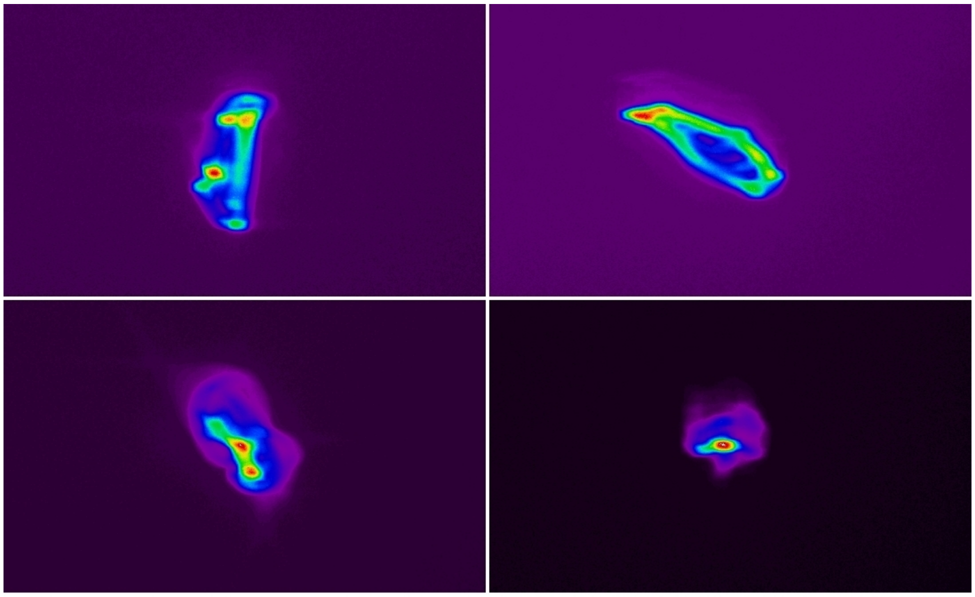


Figure 7.9: Four images of mirror spots that were taken during the H.E.S.S. Phase 2 mirror measurements at the IAAT. All of these spots passed the tests because their 80% intensity containment radius is smaller then the requirements, nevertheless, the distribution of the maxima in some of the spots is not ideal for the spot finding algorithm.

The spot finding algorithm in MACS II comes in several variants, but all based on the following idea: two images are compared by subtracting them. Spots from mirrors that have not moved disappear in the difference image, while mirrors that have moved form a pattern of negative and positive spots. The required mirror movement can then be calculated by measuring the spot movement vector from the negative to the positive pattern. Since noise is part of the images, smoothing with a Gaussian kernel or, alternatively, a grey opening filter before or after the subtraction can be applied.

From the resulting difference image, the spots are selected by setting all values close to the mean to zero ('close to' means within n standard deviations, where n is a parameter that can be changed by the user). Clusters of connected pixels are labeled as 'objects'. Pixel A is considered connected to pixel B if they are adjacent and if both have a value above zero or if both have an adjacent pixel C with a value above zero. Since this is a recursive definition, whole groups of pixels can be connected. From these labeled objects, the center of mass and the mass itself are determined and are used to order the objects. The N objects with the largest mass are considered the desired spots at the positions of their center of mass.

In the basic alignment algorithm, a single spot is searched for (N=1), which works very reliable. In the fast algorithm, the number of searched spots is bigger (N > 1), and here, a simple ordering by mass is no longer enough. Additional tests on the spot mass and possible movements of the center of mass are needed to exclude the cases in which two spots overlap.

MACS II includes the possibility to switch between the algorithm variants, since it is not

clear at the moment, how good or bad the quality of the CCD images will be once CT5 is being set up.
Last but not least, another way of determining the spot position was implemented, which is the slowest, but nevertheless very reliable: The user itself can look at two images and determine the regions where spots are missing and appearing. During the mirror alignment campaigns of the H.E.S.S. Phase 1 telescopes, the users were able to see the appearing spots in the CCD images even when the spot finding algorithm of MACS (the mirror alignment software of the first four telescopes) was not able to find them. Unfortunately MACS didn't allow them to interfere and select spot positions manually. MACSII now does. Letting the user help the software to find the spots will likely improve the speed of the overall alignment.

7.4.4 A Fast Alignment Algorithm

One disadvantage of the algorithm described in Chapter 7.4.2 is the constraint to move only a single mirror at a time (between two images taken by the Lid-CCD). One way to remove this constraint is to assume that the spot-finding algorithm can detect several spots in the image (which it theoretically can, but the weather conditions during the actual alignment render this difficult). Given that it works, there could be several mirror movements between two images. But the question then is, which spot belongs to which mirror. This can be solved by moving different combinations of mirrors at different times. For the following algorithm, all mirror spots can be initially moved out of the FoV of the Lid-CCD, but it is not strictly necessary.
For each mirror, there are (during coarse alignment) two spot positions after which the mirror is moved out of the FoV of the Lid-CCD. Each spot is connected with two events: the image at which the spot appears and the image at which the spot disappears. The second event of the first spot and the first event of the second spot are identical. When we enumerate the images, we can describe each spot with those two numbers: `spotlabel : [A, B]`.
At the same time, every mirror has a label, that consists of the numbers of the movements. The label `(0,2,5)` means that the mirror was moved before image number 0 and then again before image number 2 and 5. If we combine now the spot labels with the mirror label and take into account that the second event of the first spot and the first event of the second spot must be identical, then the spots with the labels `(0,2)` and `(2,5)` belong to the mirror with the label `(0,2,5)`. A requirement for this assignment is of course, that each spot label e.g. `(2,5)`, is unique.
Figure 7.10 shows an example from an alignment simulation of the mirror with the label `(9, 11, 12)`.

This algorithm has two main properties: the number of mirrors moved at the same time (MACS II restricts this property to 25 due to the 25 panel controllers) and the number of spots in the FoV of the Lid-CCD. The second property makes the spot-finding harder if more spots are in the FoV. More spots in the image make it more likely that some overlap,

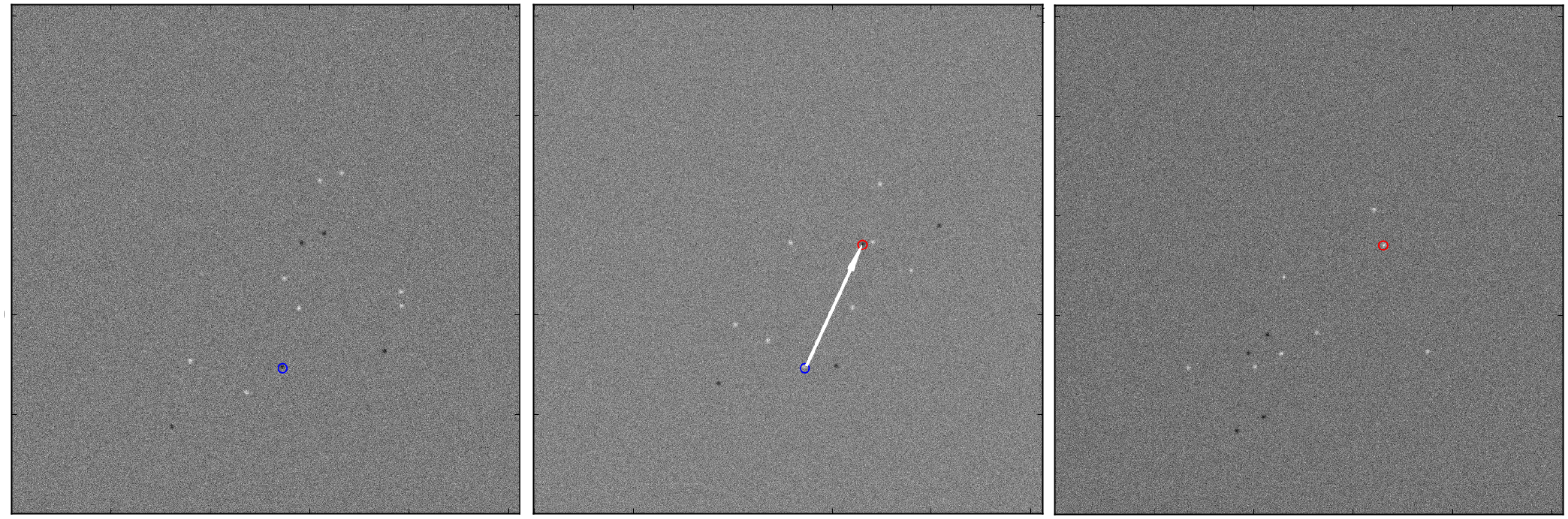


Figure 7.10: Example images from an alignment simulation with the fast algorithm. The difference images of frames 9, 11, and 12 are shown. Spots that are new on the images (compared to the images taken before) are black. While spots that had been in the images before but are no longer visible on the following images are white. Since the movement labels of the mirrors are unique, only one spot occurs in both pictures number 9 and 11 (marked blue) and only one spot occurs in both pictures 11 and 12 (marked red). These spots which are identical can then be used to calculate the transformation matrix of the mirror with the label `(9,11,12)`.

which might lead to wrongfully detected spot positions. The influence of the maximum number of mirrors moved at the same time on the number of spots in the FoV of the CCD camera can be seen in Figure 7.11.

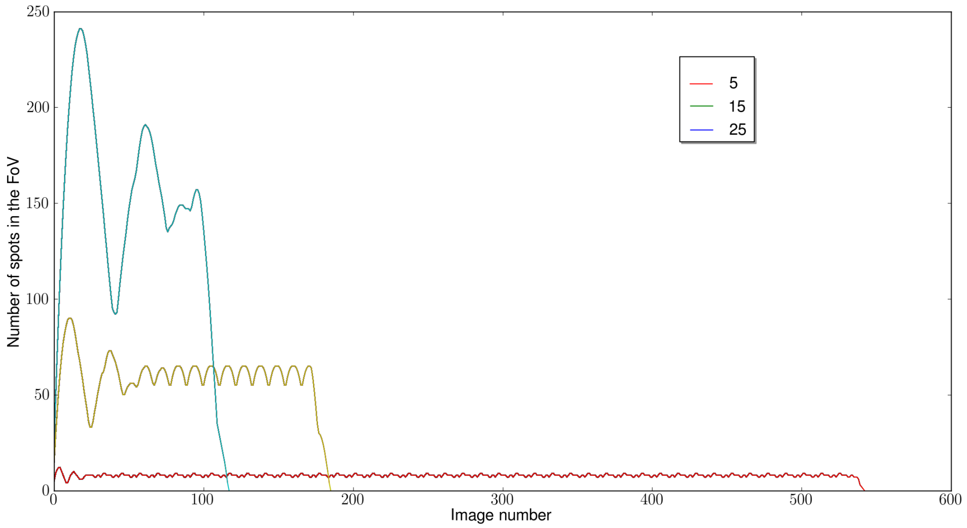


Figure 7.11: Given the maximum number of mirrors moved at the same time (5, 15, and 25), the number of mirror spots in the FoV during the alignment is shown. It can be seen that there is a highly variable initial phase, a main phase with little variation and the end phase. The higher the number of mirrors moved at the same time, the longer is the initial phase.

The current implementation takes the maximum number of mirrors that are moved at the same time as a parameter, unlike the number of spots, which can only be calculated after the algorithm has run. Assuming this number is kept fixed, we can look at the maximum number of spots that are in the FoV at the same time and the number of images that are needed until all spots are connected to a mirror. Both values can be seen in Figure 7.12 starting with moving two mirrors at the same time up to 25 mirrors.
Moving two or three mirrors both requires each 900 images, thus there is no benefit in using three instead of two. Also, when going beyond ~ 10 mirrors, the number of images does not decrease as rapidly as in the beginning. This can be understood when looking at Figure 7.11, because the initial phase, which is caused by the boundary condition of having no spots to move out of the FoV in the beginning, takes up a large part of the whole alignment. Since it is not easy to estimate how well the spot-finding algorithm will be able to handle the number of spots per image (because of weather and individual mirror spot quality), the optimal number of mirrors to move at the same time can not be determined, but will probably lie between 5 and 15.

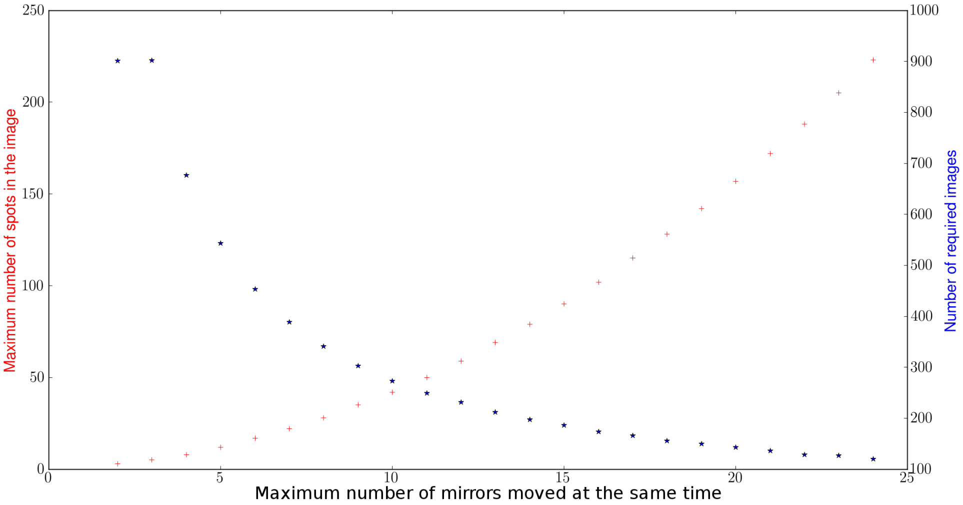


Figure 7.12: The figure shows a the maximum number of spots in the FoV of the Lid-CCD (red) as well as the number of images that are needed to identify each spot with a mirror and therefore suffices to calculate the (coarse) transformation matrix of each mirror (blue) for a given maximum number of mirrors that can be moved at the same time. The optimal number of mirrors that are moved at the same time seems to be between 5 to 15. How well the spot-finding algorithm can handle the number of spots per image depends on the weather conditions and on the individual mirror quality. Therefore, the optimal number of mirrors is hard to determine.

7.5 Results

Both, the basic and the fast algorithm, have been tested with the simulated telescope. In the initial state of the telescope simulation, each actuator was placed off the align position by a random value from a uniform distribution. At the initial state, the spots are therefore not uniformly distributed around the center position but form a parallelogram, based on the transformation matrices of the simulated mirrors. Since the movement of the mirrors on CT5 will not exactly follow a linear transformation, the mirrors will also not form exactly a parallelogram. But unlike the orientations of the actuators on CT1 – CT4, which could be mounted in 12 different orientations, the orientations of the actuators on CT5 are determined by the steel beams of the mirror support structure and are therefore the same for each mirror.

In the images below, the actuators are then moved according to the fast alignment algorithm, with up to eight mirrors moved at the same time. In Figure 7.13 the initial distribution of the spots can be seen (with 500 mirrors, not the full 890), also the 100th and 150th frame are shown. At frame 191, all spots have been moved out of the FoV of the CCD camera.

Finally, the transformation matrix of each mirror is calculated and the mirrors are moved to their calculated alignment position. This can be seen in the last image in the figure (frame 192).

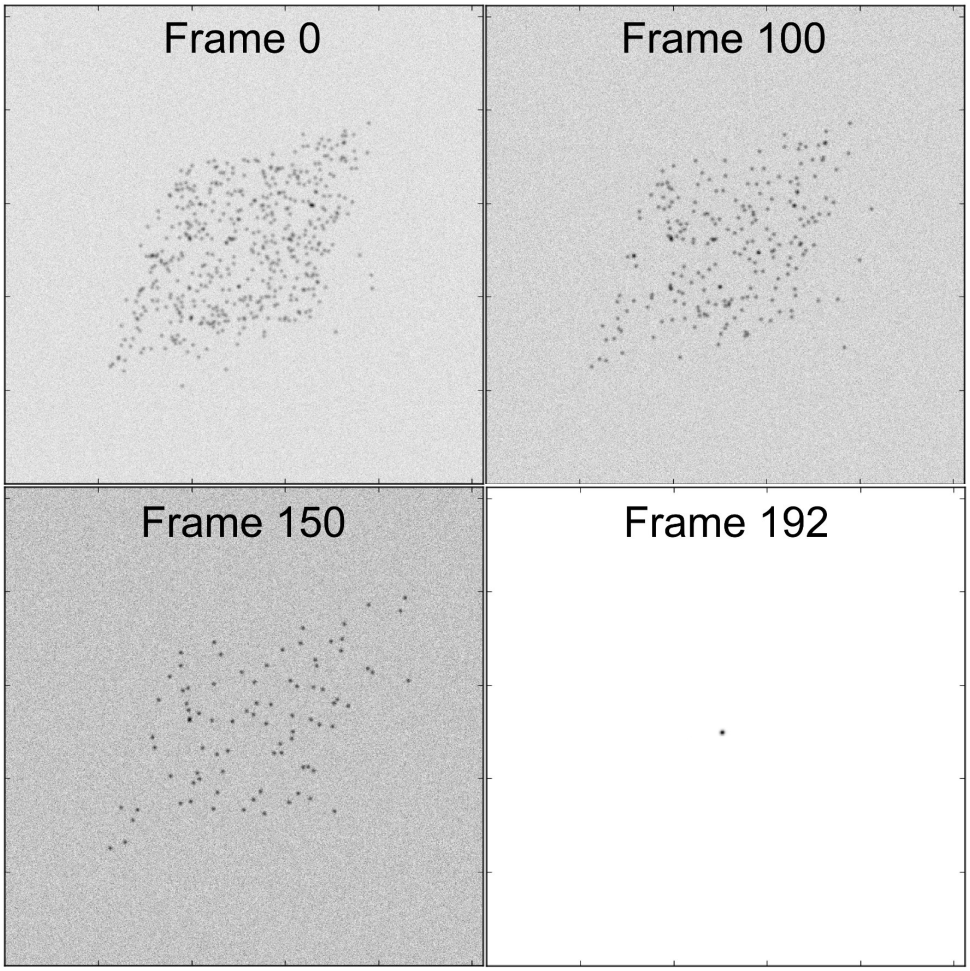


Figure 7.13: Snapshots from the simulation of CT5 via the fast alignment algorithm. The initial state with 500 mirrors is shown in frame 0 (an explanation for the shape is given in Chapter 7.5). The picture is quite chaotic as each mirror focuses on another point. In frame 100 and 150 two intermediate states are shown. There are less spots visible, because the algorithm could already determine two spots for some of the mirrors, which can be used to calculate the corresponding transformation matrix. As soon as this happens, these mirrors are moved such that the spots they cast are outside the FoV of the CCD camera. In frame 191 (which is not shown), no spot remains in the FoV – thus the frame is empty. In frame 192, the result of the successful alignment is shown: all mirrors are moved to their calculated alignment position which is now the same position for every mirror. They all focus on the same spot.

Given the simulation of CT5, the new alignment algorithm works very well and is much faster then the basic one used in H.E.S.S. Phase 1. The basic alignment procedure would have taken 1000 images and 1000 actuator movement periods, the new algorithm requires 191, thereby keeping the same alignment accuracy as the slower algorithm.

Chapter 8

The Future: an Active Mirror Control for CTA

Based on the knowledge of the mirror alignment system of H.E.S.S. and H.E.S.S. Phase 2, the IAAT takes part in the design of an Active Mirror Control (AMC) system for the Cherenkov Telescope Array (CTA). Although the exact specifications of the CTA telescopes are not yet known (they are currently being developed), one requirement for the CTA AMC seems already very clear: the AMC should be able to align many mirrors (or even all mirrors) of each telescope fast (even during observation) and continuous (that means several times per night). This can not be integrated in the current design of H.E.S.S. Phase 1 or H.E.S.S. Phase 2. A different approach was pursued by the IAAT, where the controlling logic was put directly into each motor box. This results in a much bigger number of logic components and makes the use of SoCs like the Suzaku board too expensive. Instead, micro-controllers were chosen that can communicate via CAN bus in the telescope.

From the telescope to the control room, again an Ethernet connection is planned. The connection between the telescope CAN bus and the Ethernet cable is not yet clear, as many ready-to-use components are too expensive or have been too unreliable in initial tests.

The design that is developed by the IAAT is not the only candidate for the CTA AMCs. A group from Zürich has also developed a system, which uses the wireless XBee protocol. Their design also integrates a different motor movement procedure which is able to determine the absolute position of the actuator instead of the relative movement that is used in the design by the IAAT. The absolute position measurement is achieved by using a combination of a grey code to determine the angle of the ration (within one rotation) and two hall sensors positioned at the actuator spindle (not the motor axis as in MACS II) to determine the rough actuator position (for a short description see CTA Consortium 2010).

The decision which design will be used has not yet been made, and both systems are still evolving.

There are also intentions in the CTA community to deploy an active measurement system instead of a passive star-based system. Possible solutions for such a system could be lasers at each mirror facet. But since such system have proven to be hard to handle, there is still research going on to find a suitable solution. It is clear however, that when the fast continuous mirror control is needed, a star-based system would only work, if the telescope deformations that could be compensated for by the AMC are reproducible. Then, a look-up table could be created for each zenith angle which could be used to realign the telescope. If the deformation would be dependent on more than the actual position, for example on the telescope movement that was used to get to a certain orientation or the wind conditions, a more complex look-up table would still be possible up to a certain complexity. Beyond that, an active system would be easier or more capable. Based on the experience with H.E.S.S. and H.E.S.S. Phase 2, however, I would always suggest to keep the system as simple as possible. Depending on the location(s) of CTA, extreme temperatures and weather conditions, strong UV light, and insects can be a challenge for every piece of hardware on the telescope. While some components have to be installed outside on the telescopes where they are exposed to such extreme conditions, it will be harder to protect those components if they are markedly complex. Therefore it is advisable to keep especially vulnerable components as simple as possible.

Part II

A Morphology Study of the H.E.S.S. Source J1837-069

Chapter 9

Introduction

In the second part of this thesis, an new analysis method is shown, tested and applied to the H.E.S.S. source J1837-069.
The H.E.S.S. collaboration has in the past published results on the source HESSJ 1837-069, claiming the possible existence of two separate emission regions instead of only one (see Section 9.4). Several possible counterparts in other wavelengths have been claimed by authors outside the H.E.S.S. collaboration (see Section 9.3).
The basic question to which this work tries to add additional input is, whether the spatial distribution of photons observed by H.E.S.S. can be used to claim a separate emission region. This does purposefully exclude additional, astrophysical information, e.g. from different wavelength observations. Instead, a blind search is sought-after.
After shortly looking into the basic H.E.S.S. analysis as well as into wavelet based transformations in Section 9.5.1, an analysis software called LIRA is tested for this purpose (see Section 10). Since this software, which is developed by scientists from the CHASC Astro-Statistics Collaboration, has not been used on H.E.S.S. data before, a series of tests is conducted with simulated sources. These simulated sources can be considered examples of extended, irregular shaped sources as observed by H.E.S.S. (see Section 10.6).
The tests showed that, after some problems had been identified, the software is very well suited for reconstructing extended sources in the low count regime. The LIRA software does not itself offer further tools to analyse the reconstructed images, however Connors et al. (2011) have shown ways to extend the analysis to infer significance of sub-features.
For the analysis of HESS J1837-069 (see Section 11), a feature detection, based on a white top-hat filter, separates three distinct regions. One of the regions, which is calculated with a statistical significance of 3.04σ, coincides with the previously claimed second hotspot.
The following sections give a more detailed introduction to the second part of the thesis, starting with a brief overview over VHE processes.

9.1 The Production Processes of VHE γ-rays

Thermal emission is an important process for many astronomical objects. regime of VHE astronomy however, thermal processes have a negligible effect. According to Wien's law, a black body which has its maximum emission around 100 GeV would need a temperature of about 2.5×10^{14} K, however the hottest surface temperatures of stars are in the order of $\times10^{6}$ K. Therefore other processes must be the dominant production mechanisms for VHE γ-rays. Among the possible processes are the decay of heavy particles and annihilation of dark matter. But leaving this aside, since these processes are to our current understanding not relevant in astronomical objects like Supernovae and Pulsar Wind Nebulae, there remain two main categories which result in emission of VHE γ-rays: hadronic processes and electronic processes.

Hadronic processes

If protons or nuclei are accelerated (e.g. in shock waves) and interact with ambient material, short lived hadrons are created if the energy of the primary particle is above a certain threshold. This threshold depends on the secondary hadron. Among the secondary hadrons are mesons like the π^0 and the η. The decay chain that is most important for the VHE γ-ray regime is the $\pi^0 \rightarrow \gamma\gamma$, with a threshold for π-meson production of $E_{kin} \simeq 0.3$ GeV. The π^0 are mainly produced in the reaction $pp \rightarrow \pi^0$.
A detailed calculation of the spectra created by p-p interactions in the VHE regime is given e.g. by Kelner et al. (2006). An important result of this calculation is that rough estimates of the secondary photon spectrum can be given for certain kinds of primary proton distributions. For example, power-law proton distributions (e.g. from SNRs) result in power-law photon distributions with the same spectral index, but with a maximum at lower energies. Spectral features in the primary spectrum, like an exponential cut-off, are transferred into smoother features in the γ-ray spectrum (see Kelner et al. 2006).

Leptonic processes

The main production mechanism of γ-rays that involves electrons[1] is **inverse Compton (IC) scattering** . This is a relevant process if ultrarelativistic electrons are present in a photon field of lower energies. The low energy photons can stem from the (omnipresent) cosmic microwave background but also from local, thermal radiation in the infrared or optical regime. In the scattering process between those two components, the photons gain energy and thus can reach very high energies.
The resulting photon spectrum in the VHE regime depends on the relation between the energy of the electron E_e and the energy of the photon Eγ.
In the Thompson regime (when $\xi = \frac{E_e E_\gamma}{m_e^2 c^4} \ll 1$), electrons tend to lose smaller fractions of their energy per IC event and the cross section becomes nearly independent of the energy, whereas in the Klein-Nishina regime the electrons lose a large fraction of their energy in each collision. While the energy loss rate of the electrons mainly depends on the energy density of the photon-field, the resulting spectral shape of the scattered photons is sensible to the initial electron energy distribution.
In the transition between the Thompson and the Klein-Nishina regime, the γ-ray spectra steepens: given a power-law distribution of electrons $N_e(E) \sim E_e^{-\Gamma}$, the resulting IC spectrum will have a spectral index of $\Gamma_\gamma = (\Gamma_e + 1)/2$ but will steepen by

[1] The word 'electron' is here used for both electrons and positrons unless explicitly stated otherwise.

$\Delta\Gamma_\gamma = (\Gamma_e + 1)/2$ in the KN regime (see Hinton and Hofmann 2009).

When relativistic charged particles like electrons pass though a magnetic field **synchrotron radiation** is emitted. The synchrotron spectrum ranges from the radio to the X-ray regime. Since this energy is much lower than the electron energy, the same electron population that produced the synchrotron radiation can scatter those photons by inverse Compton scattering to higher energies. This process is called inverse self-Compton scattering. The spectral energy density (SED) of a source with relativistic electrons therefore exhibits two distinct maxima. One at lower energies from the synchrotron emission and one at higher energies from the IC emission. For the case that the same electron population produced both components and the magnetic field did not change for both production regions, Aharonian et al. (1997) use the rule of thumb relation

$$E_{sync,keV} \simeq 0.07/E_{IC,TeV}B_{-5} \tag{9.1}$$

($B_{-5} = B/10^{-5}G$) between the IC and synchrotron component of relativistic electrons interacting with the CMB.

Yet another process which involves electrons and can produce γ-rays is **Bremsstrahlung**. If charged particles are deflected in an electric field, they loose energy by emitting Bremsstrahlung photons. The spectrum of the resulting photons resembles the velocity distribution of the primary particles, which is often a power law.

As long as the matter density n is low, IC processes dominate over Bremsstrahlung in the TeV regime. E.g. IC scattering on the CMB dominates in the TeV regime if $n < 240\,cm^{-3}$ (see Hinton and Hofmann 2009).

The mechanisms described above have in common that they include highly accelerated charged particles which produce γ-rays as they propagate away from the source of their acceleration. The resulting source shapes which can be resolved by IACTs are therefore often extended sources mapping the particle flow and target density. If a VHE γ-ray source is seen as a pointsource, this therefore often means that the source is very far away (extragalactic) or that the target is a compact region (see Hinton and Hofmann 2009), but also γ-ray binaries like HESS J0632+057 are seen as point-like.

9.2 The Astronomical Sources of VHE γ-rays

A number of different sources can be seen with today's IACTs. The source types are often categorized into Galactic sources and Extra-Galactic sources.

Since γ-rays in the TeV regime are nearly exclusively produced in non-thermal processes, the current understanding is that the main sources of γ-rays are related to the main sources of cosmic rays. Supernovae remnants (SNRs) are among the brightest sources in the VHE regime (Reynolds 2008), but X-ray binaries, PWNe, AGN, and star clusters are also visible to current IACTs, although in the case of star clusters it is not yet clear if the origin of the γ-rays from the direction of the star clusters is associated with genuine cluster phenomena like colliding winds. Since VHE emission traces not only cosmic rays but also target material distributions, giant molecular clouds can be seen in VHE as well.

The first Pulsar Wind Nebula (PWN) which was detected in the VHE regime was the Crab (Weekes et al. 1989). Today, about 30 VHE sources are identified as PWN and many of the unidentified sources [1] are believed to be associated with PWNs (Wilhelmi 2011).
Since this second part of the thesis will describe an alternative analysis method for H.E.S.S. data by using data of the PWN candidate HESS J 1837-069, PWNe are described in more detail now.

Pulsars and Pulsar Wind Nebulae

Pulsars are neutron stars which exhibit a pulsed lightcurve, originally detected in radio (Hewish et al. 1968), later also in X-rays and recently also in γ-rays. They are among the possible compact remnants of a core-collapse Supernova of massive stars. During a core-collapse Supernova, the magnetic field of the progenitor star is frozen and its angular momentum is conserved. The resulting neutron star can therefore be highly magnetized (in the order of 10^{12} G on the surface) and be in fast rotation (periods range from milliseconds to seconds).[2] Typical neutron stars have a radius of about 10-12 km and a mass of $1.4 - 2\,M_{\odot}$. (see e.g. Gaensler and Slane 2006).
If the magnetic field and the rotation axis of a neutron star are not aligned, the magnetic poles rotate as well. Light emission that is connected with the magnetic pole region (emission of pulsars is said to come e.g from a hot spot at the surface, but also from an emission column above the magnetic pole region) can therefore cause a pulsed signal of radiation, depending on the line of sight of the observer. This radiation is mainly in the radio, but can be high energetic as well.
VHE γ-rays originating directly from the pulsar would therefore be pulsed as well, which has been seen in the Crab (Teshima 2008, VERITAS Collaboration et al. 2011).
The origin of most pulsar related VHE emission is however more likely stemming from accelerated particles in the PWN region (see below).
The initial rotational energy of the pulsar is converted over time into other forms of energy. The converted energy rate (also known as *spin-down luminosity*) is

$$\dot{E} = 4\pi^2 I \frac{\dot{P}}{P^3} = I\dot{\Omega}\Omega \tag{9.2}$$

where P is the pulsar's spin period, I the moment of inertia, and Ω the angular frequency. The frequency change is therefore $\dot{\Omega}$, and can be described as $\dot{\Omega} = k\Omega^n$, with a constant k and the so called breaking index n. The value of the breaking index depends on the energy loss mechanism (also, the correctness of the whole ansatz itself might depend

[1] Unidentified sources in H.E.S.S. are either sources with several counterparts in other wavelength, where the origin of the γ-rays can not be clearly identified, or are sources which have not (yet) been seen in other wavelengths. For a recent review see e.g. Tibolla et al. (2009)

[2] The ATNF Pulsar Catalogue (Version 1.43), which includes rotational powered pulsars, Anomalous X-ray Pulsars (AXP) and Soft Gamma-ray Repeaters (SGR), but not accretion powered pulsars, lists pulsar periods from 1.4 ms (PSR J1748-2446ad) to 11.8 s (PSR J1841-0456) and surface magnetic fields from 6.67×10^7 G (PSR J2229+2643) to 2.06×10^{15} G (PSR J1808-2024) (Manchester et al. 2005).

on the mechanism as well). The theoretical value for pure dipole radiation is $n = 3$. Assuming this value for n and an initial spin-period much larger then the current period, the *characteristic age* τ_c of the pulsar can be deduced from the current spin-period and its derivative:

$$\tau_c = \frac{P}{2\dot{P}} \tag{9.3}$$

In pulsars, where ages[1] and spin periods are known, the characteristic age seems to overestimate the actual age (Migliazzo et al. 2002).

Pulsars emit a wind of energetic particles that collide with the SNR ejecta. These collisions can lead to a shock front, where particles can be re-accelerated. This shocked region, which expands into the surrounding medium, is called a Pulsar Wind Nebula (PWN). As already described above, the regions where energetic particles and possible target material are present are likely areas of VHE γ-ray production.

The mechanism of γ-ray production which are dominant in a pulsar/PWN complex differ for the different regions within the complex. Three main regions can be defined (see Aharonian and Bogovalov 2003):

Magnetosphere: γ-rays from the magnetosphere can be produced in several ways: curvature radiation, synchrotron radiation and inverse Compton radiation. Whether this radiation extends into TeV or has an exponential cut-off at GeV energies depends on the assumed radiation model. In the outer gap model as well as the polar cap model, the primary γ-rays are produced by curvature radiation of high energy electrons, which in turn can produce $e^{\pm}$ pairs that can initiate electromagnetic cascades. While the outer gap model (see e.g Hirotani et al. 2003) predicts a TeV component due to inverse Compton scattering, the polar cap model (see e.g Harding et al. 2002) predicts no TeV emission because the electrons are absorbed in pair-production processes within the magnetic field close to the pulsar. Since the region is close to the pulsar and related to the magnet field, this radiation is expected to be pulsed.

Unshocked Wind: Further away from the pulsar, in a region in which the pulsar wind is not yet shocked, electrons are moving with high velocities. They can (in theory) produce TeV emission via IC scattering (Aharonian and Bogovalov 2003), but are not likely to emit synchrotron radiation since they move with the magnetic field lines, which are frozen in the pulsar wind. The low energetic photons which are required for IC emission could stem from thermal radiation of the pulsar. This emission is assumed to be in the range from 10 GeV up to 10 TeV. The luminosity of this region is expected to be low compared to that of the next region.

Synchrotron Nebula: At some point, the pulsar wind interacts with the ambient matter and forms a shock-front. There, particles are thermalized and re-accelerated, e.g. through Fermi-acceleration (see e.g. Baring 2011). After leaving the shock, those particles can emit γ-rays in the TeV regime through IC scattering. The spectrum of this emission re-

[1] In Migliazzo et al. (2002), the pulsar age is estimated from its proper motion and the assumed birthplace of the pulsar (PSR B1951+32). An independent estimate on the age based on the expansion velocity of the H I shell results in a similar age estimate.

gion depends on the acceleration type. The maximum energy of the accelerated particles depends on when they can escape the region and therefore cannot be further accelerated. This energy can be written as (de Jager and Djannati-Ataï 2008):

$$E_{max} = Ze\epsilon B_s R_s \tag{9.4}$$

Here, Z is the charge number of the particles, ε is the fraction of R_s (the pulsar wind shock radius) required for the containment of the particles and B_s is the magnetic field in the outer region.

Young pulsar systems, which are at the beginning of their PWN evolution, appear point-like for H.E.S.S.. Examples for this group are the Crab (Aharonian et al. 2006a) and Kes 75 (Terrier et al. 2008). Middle aged PWN (> a few thousand years), such as Vela X (Aharonian et al. 2006b), can be seen as extended structures, often with the center of gravity of the emission showing an offset to the assumed current pulsar position. The reason for this offset can either be a high velocity of the pulsar, or inhomogeneities in the ambient matter distribution.

The VHE emission regions often seem to be far more extended than their X-ray counterparts, provided that X-ray counterparts have been detected. This is probably due to the decreasing magnetic fields, which limits the synchrotron emission regions to smaller radii. If the lifetime of the VHE emitting electrons exceeds the cooling time of the neutron star, and if the magnetic field strength is low enough, a PWN might not be detectable in the X-ray regime at all, but still be visible in the VHE regime (de Jager and Djannati-Ataï 2008).

To identify a VHE γ-ray source as a pulsar/PWN, three different ways can be pursued: firstly, the spectral properties of the source can be examined. Spectral softening of the structure as a function of the distance to the assumed pulsar is e.g. expected for PWNe (Aharonian et al. 2006).

Secondly, timing analysis can reveal emission coming directly from the pulsar (see e.g. the *pulsar trigger* technique in Aliu et al. (2008)).

Thirdly, spatial coincidence of VHE γ-ray emission with known pulsars/PWNe (based on observations in other wavelengths), can be used to identify VHE sources. The possible identification of HESSJ 1026-582 (HESS Collaboration et al. 2011) as a pulsar for example was done mainly through the spatial coincidence with the Fermi LAT pulsar PSRJ 1028-5819. The analysis method described in this part of the thesis (called LIRA, see Chapter 10) aims at increasing our understanding of the morphological properties of possibly extended sources especially in the low-count VHE regime. Apart from its direct application in understanding the source region of HESS J1837-069 (see the next chapter), the procedure might also proof helpful for identifying VHE based on their morphology.

9.3 The Source Region of HESS J1837-069

In the following sections, an analysis of H.E.S.S. data of the region around the H.E.S.S. source HESS J1837-069 (ra$18^h37^m30^s$, dec $-6°55'20''$) will be described. This data set has

previously been analysed and a detection of a significant emission from that region has been published by the H.E.S.S. collaboration (see Aharonian et al. (2005) and Marandon et al. (2008)).
During the first scan of the Galactic plane with H.E.S.S., the TeV γ-ray source HESS J1837-069 has been discovered. The initial 7.6 h of observation time resulted in a published detection significance of $13.2\,\sigma$. In the region of the TeV source, other instruments and observatories as well have discovered several sources in different wavelengths. Some of these sources have in the past been described as possible counterparts of HESS J1837-069. These findings are summarised in the next paragraphs.

AX J1838.0–0655 and AX J1837.3–0652

The X-ray source AX J1838.0–0655 was discovered with the Einstein satellite in 1988 (Hertz and Grindlay 1988) and has since then been seen by many satellites, including ASCA (*Advanced Satellite for Cosmology and Astrophysics*) and Chandra. A 70.5 ms pulsar was discovered by Gotthelf, E. V. and Halpern, J. P. (2008) (see also Gotthelf et al. 2008) in the source AX J1838.0-0655. The observation that led to this discovery was carried out with RXTE (*Rossi X-ray Timing Explorer* Bradt et al. 1993). Using Chandra data, Gotthelf et al. where able to refine the position of the pulsar in AX J1838.0–0655 to (J2000.0) $18^h38^m03.13^s\ -06°55'33.4''$ with a 1 σ error radius of ≈$0.3''$. The X-ray pulsar wind nebula surrounding it was determined to have an extension of about $r \sim 60''$. This pulsar/PWN was suggested by Gotthelf as a possible counterpart for HESSJ 1837-069, with a spin-down luminosity of $\dot{E} = 5.5$×10^{36} ergs s^{-1} and a characteristic age of 23 kyr. The pulsar has an assumed distance of 6.6 kpc, based on the association with a massive star cluster (Gotthelf, E. V. and Halpern, J. P. 2008).
In the region of HESS J1837-069 is also a second but less bright X-ray PWN candidate: AX J1837.3–0652 at $18h37m21s\,06°53'14''$. This source was detected with e.g. ASCA and Suzaku (Anada et al. 2009). No pulsation could be seen in the Chandra observations by Gotthelf, E. V. and Halpern, J. P. (2008).

RSGC1

Close to the newly discovered pulsar is a cluster of 14 red supergiants (RSGC1) (Figer et al. 2006), which was the largest of its kind in the Galaxy at the time of the discovery. The cluster mass was estimated to $2*10^4$ $M_\odot$ and its age to 7–12 Myr. The distance of the cluster was estimated to $5.8\,kpc$. Figer et al. claim that this cluster might be associated with HESS J1837-069 through the possibility of supernova activity in the cluster. This claimed supernova activity is mainly based on the cluster age, with one object, GPSR5 25.252-0.139, likely being a crab-like SNR.

9.4 Motivation for further analysis

In Marandon et al. (2008) the southern part of the TeV emission region of HESSJ 1837-069 was claimed to be a possible second source with a peak detection significance of $\sim 11\sigma$. To establish this second source, a slice through the H.E.S.S. excess map was created which included the southern as well as the northern part with a width of $0.2°$. In the published results (see Figure 9.1) it can be seen that the existence of a second source is very likely, although it must be noted that there seems to be an overlap between the sources that leads to no clear spatial separation. Whether this overlap is real in 3D, an artifact of the 2D projection, or the 1D slicing technique is of course not clear from this plot. To ensure that a real separation of the sources is possible (that the emission goes to zero between the sources) the influence of the PSF has to removed from the image. This process is often called deconvolution and is part of the image reconstruction shown later in this thesis.

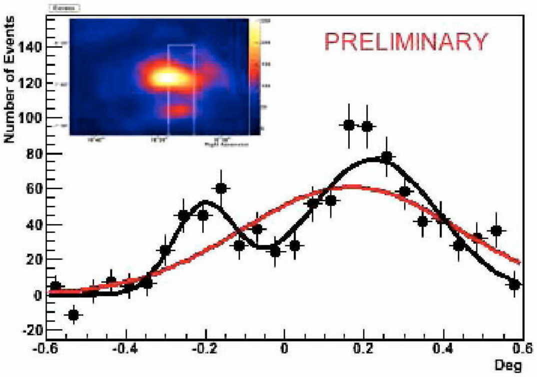


Figure 9.1: The figure is taken from Marandon et al. (2008). A smoothed excess map is shown in the upper part along with a rectangular region that was projected along its long axis to provide the 'profile' of the sources used in the lower part of the image. The two fitted functions, a Gaussian shaped function and the combination of two Gaussians, show the possibility of a second source.

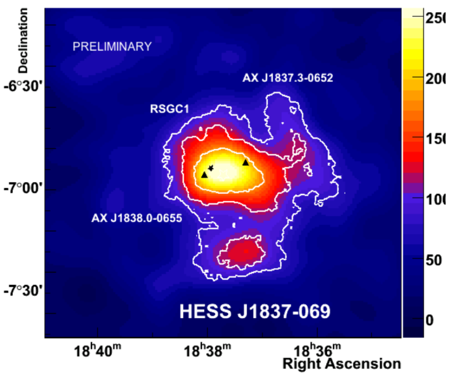


Figure 9.2: Taken from Marandon et al. (2008):'Excess map smoothed with the PSF (~4′). Significance contours in white (for 7, 10, 15 σ). The two triangles show the positions of the X-ray sources AX J1838.0-0655 andAXJ1837.3-0652.[...] The star represents, the Star Cluster RSGC1 situated at 6.6kpc'.

There are several ways to explore the possibility of several separated sources instead of a single larger source. The most basic versions, however, have to make assumptions about the source shape, e.g. point-sources or simple geometries like 2D-Gaussians. In those cases, a least-square fit of one or more 2D-Gaussians is conducted and the reduced χ^2 values of the different fits are being compared to each other.

In the procedure used in Figure 9.1 however, the selected image region was projected along the right ascension axis into 26 bins[1] and then fitted by a single 1D-Gaussian func-

[1] This procedure is a simple re-binning with bin size in on dimension equal to the slice width and the bin size in the other dimension equal to an arbitrary value which provides sufficient statistics.

tion and a combination of two 1D-Gaussian functions. The reason for choosing this procedure was mainly the low count rate which made it difficult to fit 2D-Gaussians or other shapes to the image of the source region directly: Least-square fitting and significance testing via χ^2 methods are not possible since both rely on a normal distribution of the parameters, while other popular methods like maximum likelihood fits in combination with the F-test or the likelihood ratio test (LRT) cannot be used, since in F-test and LRT, 'the null values of the additional parameters may not be on the boundary of the set of possible parameter values' (Protassov et al. 2002). This boundary is given by the fact that we are dealing with a counting statistic and negative values are not allowed/possible, but zero is the null value for an additional source component and absorption is excluded by restricting the fit parameters to positive values.

The slicing and re-binning procedure used here is often used in H.E.S.S. analyses, for example also in HESS Collaboration et al. (2011) to establish the two sources interpretation. It must be noted, that the data which were used for this analysis of HESS J1837-069, the excess map (see Section), is not the optimal choice for such a fit, because the bins of the map are correlated. The background estimation technique, the ring background maker, leads to a correlation of the excess value of bins which are not too far apart. The correlation distance depends on the ring radius which is used for the background estimation and on the size and location of so called exclusion regions, which are supposed to contain sources and are therefore excluded from the background estimation. This correlation leads to difficulties in the error estimation and complicates the fitting procedure.

A better alternative would be to base source fits on the true observation counts (so called 'on' counts) and fit the background at the same time as the source shape(s). Furthermore, the assumption of a Gaussian shape for both sources is only founded on the simplicity of the model and can not be used to reveal e.g. arc-like structures. Extended emission at the Galactic center ridge (Aharonian et al. 2006) for example, which coincides with molecular clouds, are not best described by Gaussian shapes.

Both the possible association with several counterparts, and the assumption of a second hot spot or at least a second emission component lead to the need for a finer grained spatial analysis of the region which is also suited for the low-count nature of the H.E.S.S. observation and which does not assume simple parametrised source shapes.

9.5 The H.E.S.S. Data Analysis

The short description of the source region already used several terms that refer to analysis products stemming from the H.E.S.S. Analysis Program (HAP). Those products will be introduced in more detail in the following.

Observation

The image which is used as observation data in the LIRA analysis (see Section 10) is extracted by the HAP (version: 10-06) from 32.58 h of observation of the source region.

The 75 individual observations of HESSJ 1837-069 (called ‘runs’ in H.E.S.S., each with a duration of ~ 28 min) were taken between May 2004 and June 2007. The maximum offset allowed for events in this analysis was 3°. The selection of the γ-like events was done with hard-cuts. The events are binned into an image with a pixel size of 0.02° in right ascension and declination.[1]

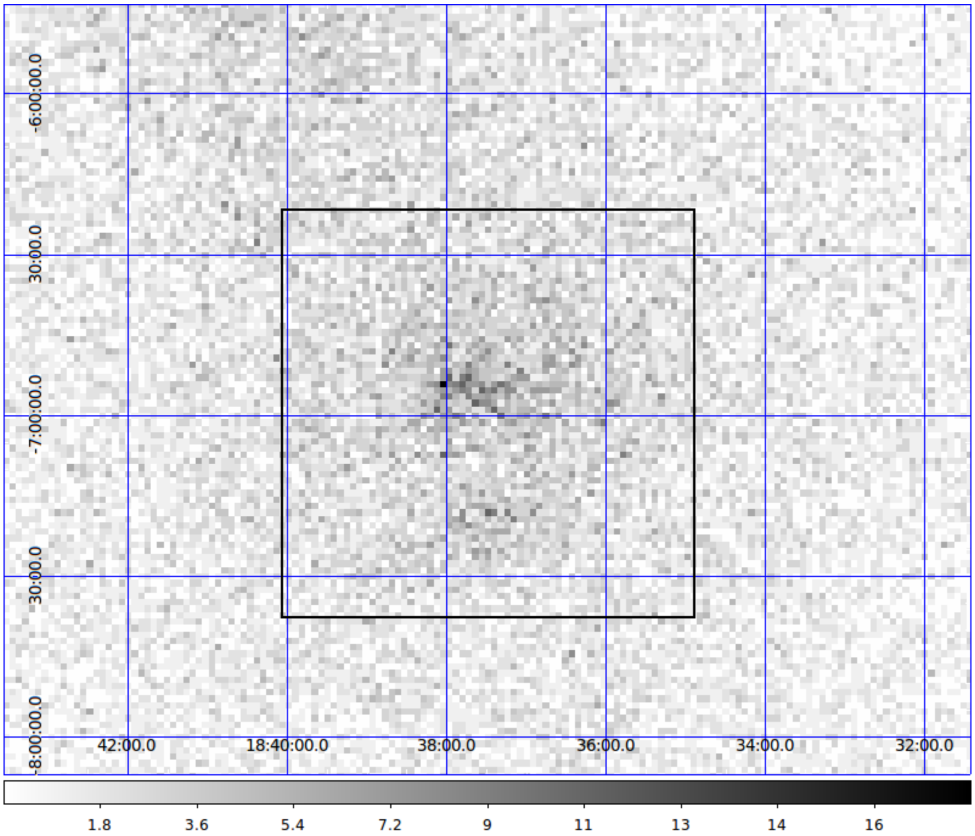


Figure 9.3: HAP ‘on’ counts from 32 h of observation. The rectangle shows the region which was used for the LIRA analysis.

PSF and Exposure

The PSF of Cherenkov telescopes like H.E.S.S. depends on many parameters.
Firstly, only the PSF of γ-like events is considered, although in reality, the ‘on’ events can be either gamma events or cosmic ray events which passed the selection cuts.
Secondly, the PSF depends on the energy of the photons, therefore a distribution of energies is assumed based on known or assumed spectral properties of the observed sources.

[1] HAP skyhists (images) use the Plate Carrée projection, the CAR keyword in the world coordinate system (WCS) (Calabretta and Greisen 2002). One property of this projection is that meridians, parallels, and poles are equally spaced, parallel, straight lines. Indicatrices (infinitesimally small circles of equal size on the globe) are ellipses (except for one meridian, typically the equator) of varying size and shape. The compromise made by HAP is to create a pixel width and height where the meridian of circular indicatrices goes through the center of the skyhist.

The selection cuts which are applied during the analysis play an important role for the same reason, since they change the distribution of energies. Comparing the PSF of different cuts, it can be seen that hard cuts give the best (most localized) PSF. These restrict the shower amplitude to more than 200 pe, the mean reduced scaled width (MRSW) to under 0.7 and the offset Θ^2 to 0.01°, which reduces the number of background events by a factor 7 (Funk 2005). These cuts are therefore used to analyse the morphology of sources. Thirdly, the atmosphere itself plays an important role in the correct detection and reconstruction of the events, therefore the zenith angle of the observation is an important parameter as well.

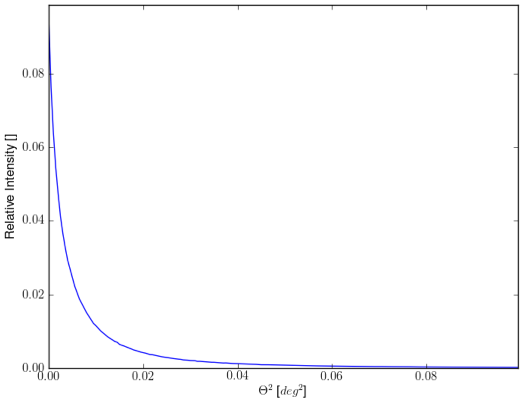


Figure 9.4: HAP PSF for the observation of HESSJ 1837-069. A 2D version of this PSF was used as input for the LIRA analysis.

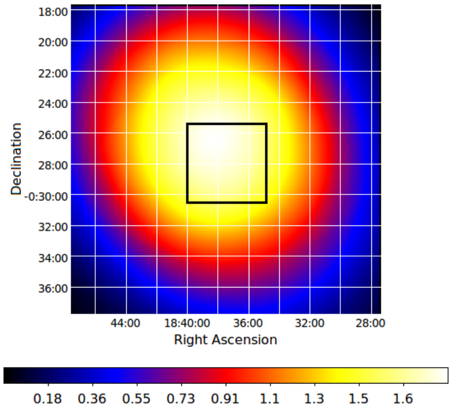


Figure 9.5: HAP exposure map. The rectangle shows the region that was used for the LIRA analysis.

For each observation, an individual PSF is calculated by HAP. The PSF which is used in the LIRA analysis is directly taken from the HAP analysis software. HAP creates a PSF based on simulations and the observation parameters. The input configuration to this analysis are so called 'hard-cuts', which are optimized for sources with a spectral index of $\Gamma = 2.0$ (Aharonian et al. 2006a).

The PSF can be well approximated by a combination of two Gaussians, one sharp and one broad, the second accounting for badly reconstructed events. The PSF can be seen in Figure 9.4. The exposure map (like the PSF) in Figure 9.5 was directly exported from the HAP. The rectangle in the exposure map marks the region which was used for the LIRA analysis (see Section 10).

Excess Maps

Although many background events are already ‘cut away’ by the shower parameter cuts, the ‘on’ image still consists of source events and background events, mostly from hadronic cosmic ray showers and at low energies also from cosmic ray electrons.

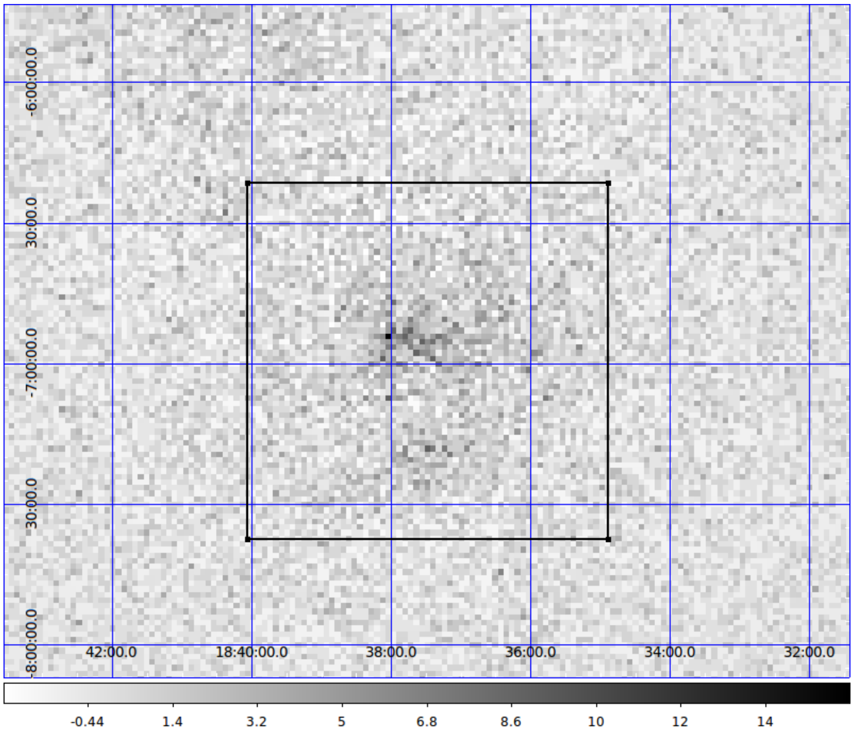


Figure 9.6: HAP excess map. The background was determined with the ring background maker (Berge et al. 2007) with a ring of $0.5°$ and an area factor of 7 ($\Omega_{\text{off}} = 7 * \Omega_{\text{on}}$). The rectangle shows the region which corresponds to the region analyzed with LIRA (see Section 10), the excess map, however, was not used. Instead, the LIRA analysis uses its own system to determine the background.

If the number of events in the region of interest is N_{on} and if the assumed number of events which only stem from the background is N_{off} (measured in the ‘off’-region), the number of true source events in the ‘on’ region can be calculated as:

$$N_{\text{excess}} = N_{\text{on}} - \alpha N_{\text{off}} \tag{9.5}$$

with the normalization factor α is thereby mainly (with a small correction for different acceptances in the different regions) defined as the ratio of the solid angles of the region of interest and the ring region $\alpha = \Omega_{\text{on}}/\Omega_{\text{off}}$ or with the full definition:

$$\alpha = \frac{\int_{\text{on}} \epsilon^{\gamma}_{\text{on}}(\Theta_{x,y}, \phi_z)\, d\Theta_{x,y}, \phi_z}{\int_{\text{off}} \epsilon^{\gamma}_{\text{off}}(\Theta_{x,y}, \phi_z)\, d\Theta_{x,y}, \phi_z} \tag{9.6}$$

Here ϵ^{γ} is the acceptance of H.E.S.S. for γ-like background events, which depends on the zenith angle ϕ_z and the position in the field of view ($\Theta_{x,y}$). The resulting image from a HAP analysis can be seen in Figure 9.6. The value for the background N_{off} was determined by the so called *ring background maker.* In this kind of background determination, an annulus around the region of interest is drawn, in which all counts are summed up (N_{off}). In the excess map in Figure 9.6 a ring of $0.5°$ and an $\Omega_{\text{off}} = 7 * \Omega_{\text{on}}$ was used.
Equation 9.5 is often associated with counts coming from sources (e.g. Li and Ma 1983). This is however not the case: the definition of the excess counts can only describe source counts when both N_{on} and N_{off} follow a normal distribution (or have large enough values to be approximated by a normal distribution). In general however, the excess map in the case of low pixel counts does not describe a source and can only be used as a measure of distance between the 'off' and the 'on' counts.

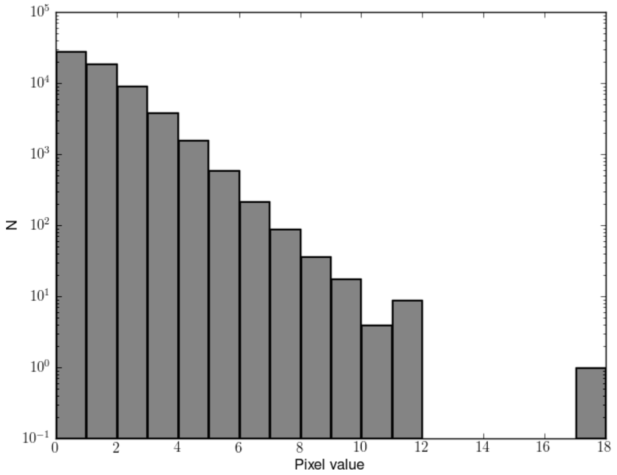


Figure 9.7: Histogram of the pixel values of the 'on' map (see Figure 9.3).

In Figure 9.7, the histogram of the pixel values in the 'on' map (Figure 9.3) can be seen. With the maximum at 18 counts and the majority of pixels having a value well below 10, the assumption that the variates $O_{(x,y)}$ of the counts in each pixel follow a normal distribution seems not to be supported. Therefore the excess map can not be interpreted as a source map. A detailed discussion about the problem of analysing 'on/off' measurements in the low-count regime is given by Loredo (1992).
In order to get an image which can be interpreted as showing a source structure, the pixels values should be better described by binomial or Poisson distributions, or the images have to be transformed by increasing the binsize: If the binsize is larger, there are more counts per bin, and at some point, the underlying distribution can be assumed normal. The obvious disadvantage of this procedure is that spatial information is lost.
LIRA (see Chapter 10) uses an alternative background determination method which takes the low-count nature of VHE observations fully in account.

Significance Map

Another interesting output of the HAP analysis are *significance maps*. Significance in HAP is calculated based on a formula by Li and Ma (1983):

$$S = (-2\ln\lambda)^{1/2} = \sqrt{2}\{N_{\text{on}} \ln\left[\frac{1+\alpha}{\alpha}\left(\frac{N_{\text{on}}}{N_{\text{on}}+N_{\text{off}}}\right)\right] + \\ N_{\text{off}} \ln\left[(1+\alpha)\left(\frac{N_{\text{off}}}{N_{\text{on}}+N_{\text{off}}}\right)\right]\}^{1/2} \tag{9.7}$$

This defines the significance of a number of counts (N_{on}) under the assumption that they could also be a random fluctuation of the background (αN_{off}). This definition of significance could also be called a *detection significance*, since it determines whether there is a source at all in the region which was used to determine N_{on}.

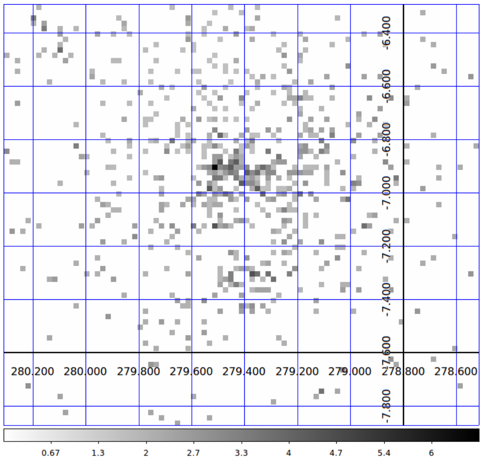


Figure 9.8: HAP significance map of the region of HESSJ 1837-069.

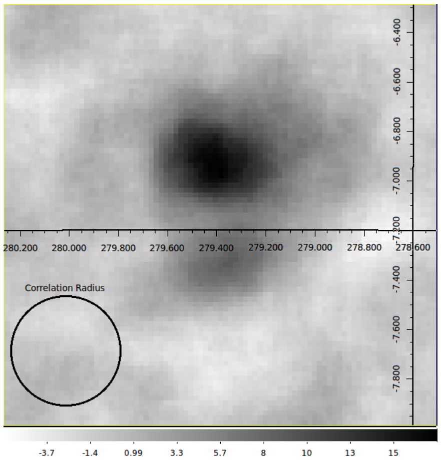


Figure 9.9: HAP correlated significance map. A correlation radius of 0.22° was used.

The significance map calculated by Equation 9.7 of HESS J1837-069 can be seen in Figure 9.8. What is apparent is the lack of values ≥ 0 in large parts of the image (shown in white). This can easily be understood because Equation 9.7 does not define a significance if the number of 'on' counts in the pixel is zero. In fact, Li and Ma (1983) have written in their original paper that the equation is only valid for $N_{\text{on}} \gtrsim 10$ and also $N_{\text{off}} \gtrsim 10$. The equation was tested by the authors by making Monte-Carlo simulations for different values of 'on' and 'off' counts. Counts fewer than 10 were not tested and will in general not work, because one assumption of Equation 9.7 is, that $-2ln\lambda$ follows a χ^2 distribution of one degree of freedom. Which means that in general, the equation is only valid in the regime where the Poisson distribution of the counts can be approximated by a normal distribution.

This discussion dealt with significance maps like Figure 9.8, which could be called *pixel-wise significance*. Because the counts in the shown images are too few to actually use Equation 9.7 for most parts of the image, the pixel-wise significance is not used. Instead a 'correlated significance map' is produced, which can be seen in Figure 9.9 for a correlation radius of $0.22°$ (which is also shown in the image). The term 'correlated map' is used here, because it is common in the H.E.S.S. collaboration. What is meant by that term is that the individual pixel values are no longer independent, because some mathematical procedure was used that somehow connected several pixel values. In the case shown here, the 'correlated significance map' is 'correlated' in several ways: firstly, the background (N_{off}) is calculated with the ring background maker, which uses overlapping image regions to determine the background for neighbouring pixels and secondly, the number of excess events used for Equation 9.7 is taken from a top-hat function, a circle with the 'correlation radius'. This has several effects: the first is that more counts are taken into account and therefore Equation 9.7 becomes valid again. The second is that no longer a pixel-wise significance is created but something closer to a source-wise significance under the assumption of a circular source shape of the size of the correlation radius. The third, however, is that the origin of the counts can no longer be assumed to belong in this case e.g. to the southern source only. The image is the result of a convolution of the true source distribution with the PSF (through the observation process) and a circle with the correlation radius (through the analysis procedure). This last effect leads to a higher significance in the southern part than would be expected if only a single source with the intensity of the southern part would be present in the map. In Figure 11.12 this effect can be seen under the assumption that the source reconstruction described there is valid. This estimation shows that about 122 counts would be stemming from regions surrounding the actual source, which itself contributes about 240 counts.

Since this a known problem with the HAP significance maps, the existence of a second source in the region of HESSJ 1837-069 relies on other means like the slicing technique which was shown at the beginning of this chapter.

In Chapter 10.3, 10.4, and 10.5 alternative means of assessing the significance, in particular a source-wise significance which only takes the southern part of the emission into account, will be shown.

The procedure used in this thesis is the following: separating the signal from the noise, deconvolving the signal to remove the effect of the PSF to enable the third step: separating the signals of individual, separate sources/source regions. Since no clear assumption on the shape of the sources can be established (point sources, arc-like structure, arbitrarily shaped extended features are all – in principle – possible), the separation is not based on 2D-Gaussians, or similar parametrized models, but it uses pure background pixels to separate sources. This simple procedure requires the true/deconvolved source distribution, which would not be needed if parametrized models would be assumed for the shapes of the sources.

LIRA provides a means to accomplish the first two steps. The basis of LIRA is a wavelet like representation, which will be introduced in the following section.

9.5.1 Wavelets

Fourier Transform

A transform of a signal, such as the Fourier- or the wavelet transform, converts the information stored in an image or a series of measurements from one representation (often a spatial representation) into another representation (for example a frequency representation) without losing information. A typical example of a transform which is often used in physics is the Fourier transform:

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(t)\, e^{-2\pi i t \xi}\, dt, \tag{9.8}$$

The information which was represented as a function of time $f(t)$ is now expressed as a function of the frequency $\hat{f}(\xi)$. The reason for using such transforms lies in the properties of the different representation. One often looked-for property is that the representation is *sparse*. Following Starck et al. (2010), a finite signal vector $x = [x[1]...x[N]]$ (such as an image or a timing measurement) is sparse, if most of its entries are zero. A *k*-sparse signal is defined as having k entries with non-zero value. In the case of the Fourier transform, a purely sinusoidal signal defined at n time-bins would require n values in the time domain, but could be expressed by just 1 number in the frequency domain, thus it can be called *1-sparse*.
Another desirable property is faster calculation time: the calculation time of a convolution of two images with N pixels is proportional to N^2 (in information theory, the algorithm is said to be O(N^2), see for example Knuth 1976), whereas the combination of the Fourier transform, a multiplication of the two transformed images and an inverse Fourier transform is proportional to $N * logN$ (which is said to be O($N * logN$)).

From Fourier to Wavelet

The Fourier transform is perfect for stationary signals, because the analysing functions are sine and cosine waves, which have time-independent frequency content. But most interesting signals like real astronomical images are not sparse in their frequency representation, because they exhibit local structures. Thus, a transform which is much better suited for these signals than the Fourier transform is the wavelet transform. This transform is a way to represent the signal in a localized manner. In the example of Equation 9.8, the transformed representation would be a function of the time and the frequency.
While the analyzing function in the Fourier transform is a wave (a periodic function with one frequency) the analyzing function in the wavelet transform is a wavelet. Figure 9.10 shows both a wave and two different wavelet representations.

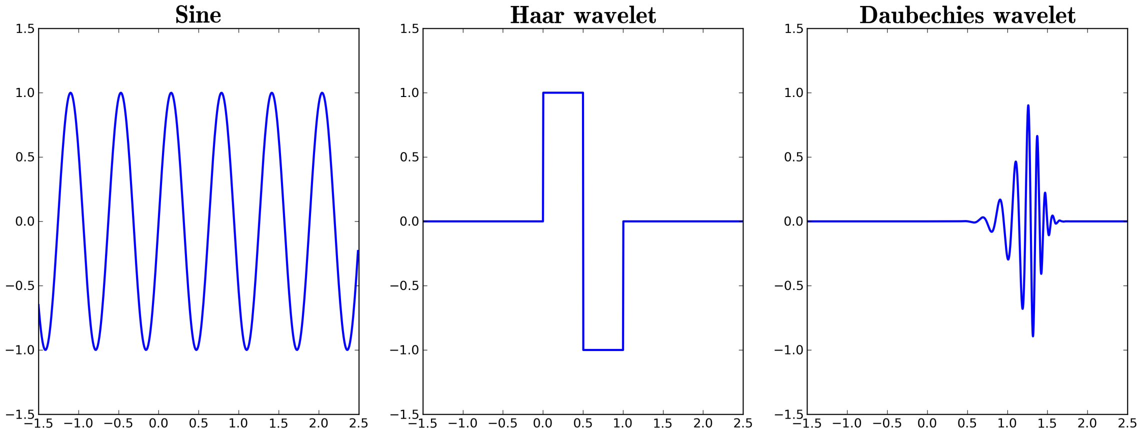


Figure 9.10: Example of a wave and two different wavelets.

Wavelet Properties

A function $\Psi \in L^2(\mathbb{R})$ can be called a wavelet, if it has a zero mean,

$$\int_{-\infty}^{\infty} \Psi(t)dt = 0 \tag{9.9}$$

if it is normalized,

$$\|\Psi\| = 1 \tag{9.10}$$

and if it is centered in the neighbourhood of $t = 0$. With such a wavelet, a basis for the wavelet representation can be formed. Often such a basis is called a *dictionary* of *atoms* (e.g. Starck et al. 2010, Mallat 2008). The atoms are single signal representation templates. The dictionary is an indexed collection of atoms. In the case of a classical image, the atoms can be seen as Kronecker deltas, in the case of Fourier dictionaries, the atoms would be frequencies. For the wavelet transform, the atoms are created from wavelets by scaling Ψ by s and translating Ψ by u:

$$\mathcal{D} = \left\{ \Psi_{u,s}(t) = \frac{1}{\sqrt{s}} \Psi\left(\frac{t-u}{s}\right) \right\}_{u\in\mathbb{R},s\in\mathbb{R}^+\ \{0\}} \tag{9.11}$$

A dictionary of this kind forms an orthonormal basis. One of the first examples of a wavelet was described by Haar (1910), see Fig. 9.10:

$$\Psi(t) = \begin{cases} 1 & : \text{if } 0 \leq t < 1/2 \\ -1 & : \text{if } 1/2 \leq t < 1 \\ 0 & : \text{otherwise} \end{cases} \tag{9.12}$$

Continuous Wavelet Transform

The continuous wavelet transform (CWT) of a one dimensional function $f(x) \in L^2(\mathbb{R})$ can be written as (Starck et al. 2010):

$$W(u,s) = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} f(x) \Psi^* \left(\frac{t-u}{s} \right) dt \tag{9.13}$$

Where s is called the scale and u is a positional parameter and $\Psi(t)$ is the analyzing wavelet or mother wavelet and Ψ^* is its complex conjugate. The wavelets in Fig 9.10 show Ψ, the analysing wavelet. The original function can be recovered via the following transform:

$$f(t) = \int_0^{\infty} \int_{-\infty}^{\infty} \frac{1}{s^2} W(u,s) \frac{1}{\sqrt{|(s)|}} \tilde{\Psi} \left(\frac{t-u}{s} \right) ds\, du \tag{9.14}$$

Here $\tilde{\Psi}(t)$ is the dual function of $\Psi(t)$.
The continuous wavelet transform is the basis for related discrete transforms that are very common in modern signal analysis like the discrete wavelet transform (DWT) and the stationary wavelet transform (SWT, also called *algorithme à trous*).

Stationary Wavelet Transform

A detailed discussion of the stationary wavelet transform in astronomy is given in Starck and Murtagh (2006). The main part of this book is dedicated to this analysis method and a number of impressive examples with astronomical images are given. Together, they show the capabilities of this method when properly combined with dedicated noise removal techniques.
The stationary wavelet transform produces wavelet coefficients W_j at each scale. In the case of the DWT, these coefficients form a pyramid, where the number of coefficients increases with the scale until the number of coefficients is the same as the number of pixels. In the case of the stationary wavelet transform though, the number of coefficients at each level is the same, which allows to study locations in the image without scaling the sub-scales: The original image Λ can be written as the sum of the wavelet coefficients and the last smoothed image $c_{K,i}$:

$$\Lambda_i = c_{K,i} + \sum_{k=1}^{K} W_{k,i} \tag{9.15}$$

The result of such a stationary wavelet transform of the region of HESS J1837-069 can be seen in Figure 9.11.

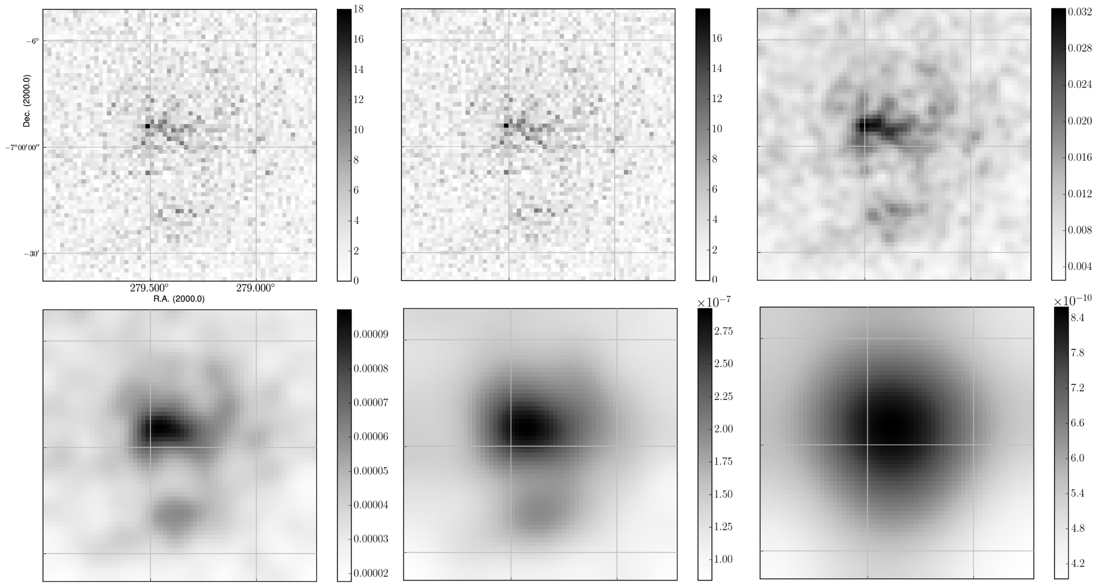


Figure 9.11: Stationary wavelet transform of the 'on' map of the region around HESS J1837-069. The first image is the original, second to fifths image are the scales, and the last image is the last smoothed image.

This wavelet representation allows to see features at different scales without the broader or smaller features overlayed. This in itself is an interesting property. But the next step in a multiresolution analysis (MRA) would then be, to remove the coefficients at each scale which are below the noise level at that scale.

The original, de-noised image could be reconstructed from the remaining coefficients, summing the signals at each scale. This is different from removing noise per pixel in the whole image, like background subtraction, because weak, extended emission which is below a significance level in each pixel, will still be visible in the reconstructed image. This problem is illustrated in Figure 9.12.

Unfortunately, selecting coefficients which are due to background noise is not always easy. In the case of normal distributed noise, the coefficients can directly be rejected, based on their intra-scale distribution, at levels $k\,\sigma$. The absolute value of the coefficients is important, because negative coefficients can also be significant. If the noise does not follow a normal distribution, Starck and Murtagh propose to apply an Anscombe transform to the data in order to be able to still treat the coefficients as if they arose from normal distributed noise. If this is not possible because the number of counts is too low for the Anscombe transform, the procedure described in Starck and Murtagh (2006) gets more complicated: it is not directly accessible, how a certain noise level will show up in the different scales.

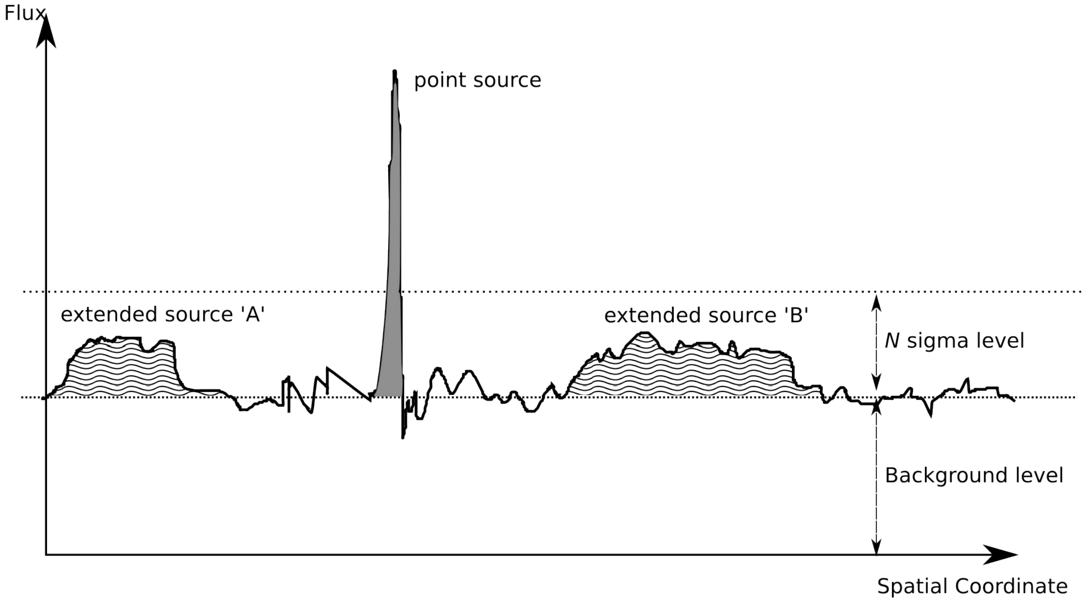


Figure 9.12: The difficulty of detecting extended features with common sigma-level based detection algorithms is exemplified (graphic based on Starck and Murtagh 2006).

Therefore the outcome of different levels of noise must be simulated for the specific wavelet family which is used in the wavelet decomposition. This simulated treatment includes histogram auto-convolution and although described in Starck and Murtagh (2006), it is not completely clear how reliable this approach is. The described approach was also studied in an H.E.S.S. internal note (Marshall 2007). Here, Jesse Marshall comes to the conclusion that the information given in the book is not sufficient to implement this algorithm.
Other multiresolution techniques like the Haar based TIPSH algorithm (Translation Invariant Poisson Smoothing using Haar wavelets) are also studied in this note with the conclusion that “The inability of the wavelet transform to discuss exact object-wise significance is a major disappointment”.

9.5.2 Multiscale likelihood analysis

Wavelet analyses are deeply connected with mulitresolution analysis (see for example Daubechies (1992) and Starck and Murtagh (2006)). According to Kolaczyk and Nowak (2004), multiresolution analysis features three important properties:

- a hierarchy of nested sub-spaces
- an orthonormal basis

- a scaling between and translation within sub-spaces

These properties ensure that a function $f \in L^2(\mathbb{R})$ can be decomposed into a dictionary of atoms (e.g. the wavelet coefficients and the corresponding wavelet functions) which contain the same information as the original function but separated and localized into combinations of scale and position.

In analogy to this set of properties, the *Likelihood Multiresolution Analysis* can be described by a similar set of properties that take the same functions as the properties for wavelet based multiresolution analysis, but with one added property to fully replace the orthogonal basis condition (see Kolaczyk and Nowak 2004). Accordingly, they list the following conditions:

- A hierarchy of recursive partitions $\mathcal{P}$
- The independence of the pixel values X_i (e.g. not preprocessed in a way that introduces a correlation), and the L-independence of the components Θ [1] with respect to the likelihood of X_i:

  $$p(X|\Theta) = \prod_{i=0}^{N-1} p(X_i|\Theta_i) \tag{9.16}$$

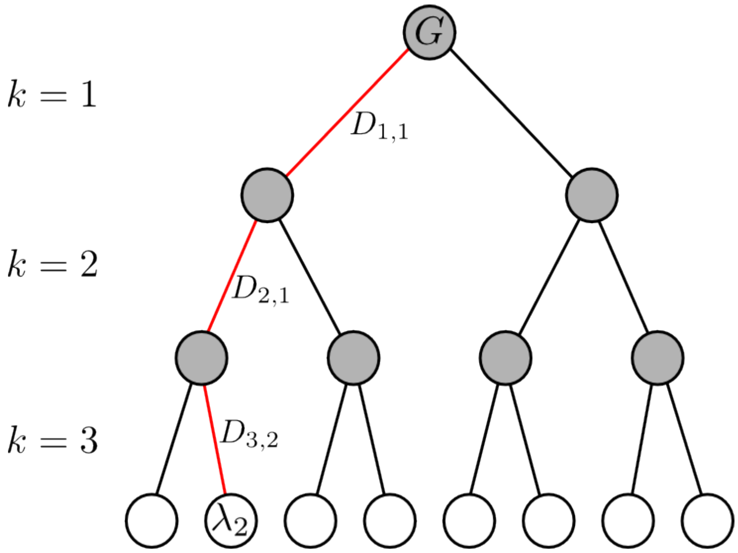
  where N is the number of resolutions. This results in an image width and height of 2^N. This property resembles the orthogonal basis in the wavelet case.
- The reproducibility between partitions
  This means that $p(X_i|\Theta_i)$ must follow a distribution which is invariant under summation (Kolaczyk and Nowak 2004). The authors show that this and the next condition hold for the family of Gaussian, Poisson and multinomial distributions.
- The decoupling of parameters between partitions
  This property is needed to ensure that only local information is stored in Θ_i.

Such a multiscale representation of an image with pixel values Λ_i can be written as:

$$\Lambda_i = G \prod_{k=1}^{K} D_{k,l_{k(i)},m_{k(i)}} \tag{9.17}$$

where G is the sum of the intensities in all pixels, and $D_{k,l_{k(i)},m_{k(i)}}$ are the split proportions at scale k corresponding to the group of pixels containing pixel i.

[1] For detailed definitions of these conditions, see Kolaczyk and Nowak (2004)



$$\lambda_i = G \prod_{k=1}^{K} D_{k,m_{k(i)}}$$

Figure 9.13: Visualization of the likelihood multiscale representation of an image. G is the sum of all pixel values. $D_{k,m}$ are the split probabilities. LIRA uses this representation with four-way-splits instead of the two-way-splits which are shown in this image. The pixels of the image are represented by the white dots in the lowest row. An example for pixel 2 is highlighted: $\lambda_2 = G * D_{1,1} * D_{2,1} * D_{3,2}$.

The resulting structure can be thought of as a pyramid or a tree, like e.g. the coefficients of the discrete wavelet transform, but unlike the stationary wavelet transform, which produces at each scale the same amount of pixels. This structure can be seen in Figure 9.13 in a simplified manner. The simplification affects the number of splits on each level: pixel values are split into two parts and not into four parts like it is done for 2D data in LIRA. Figure 9.13 therefore describes a 1D version of the LIRA multiscale representation.

The image with its full resolution is at the bottom of the pyramid. Since the algorithm only allows the same number of splits (two in Figure 9.13, four in a 2D image), the number of pixels in the image can only be 2^nx2^n. A possible way to bypass this restriction would be to embed the image into a larger image that conforms to this restriction. The LIRA analysis software does however not use such a solution.

The NK (Nowak and Kolaczyk) Model

The multiresolution representation is also called NK model, since important aspects of the representation have been published by Nowak and Kolaczyk (see e.g. Kolaczyk and Nowak 2000; 2004). In these papers, the multiscale representation is used in connection with an expectation maximization algorithm to receive a best-fit (the maximum likelihood estimator). The algorithm described there is also implemented as an alternative algorithm in LIRA.

A serious shortcoming of most wavelet analyses as well as of this newer factorization analysis is, that it is not easy to understand the statistical significance of the (wavelet) coefficients or the error on the best-fit image.

In the following section, an analysis method (LIRA) will be described, which uses the same multiscale representation, but in conjunction with a Bayesian Monte-carlo method. This ensures that one does not only get a best-fit, but also measures of uncertainty of the fitted image.

Chapter 10

The Low-count Image Restoration Analysis (LIRA)

In the previous chapters it was shown how the analysis of H.E.S.S. data could be improved. Especially morphology analysis in the low-count regime could profit from more advanced techniques. Standard HAP products contain significance maps, but not in a way that allows to access the localized significance of substructures of possible sources. Alternative analysis methods have been explored for H.E.S.S. data in the past, some of which include deconvolution techniques.

LIRA could be one of these analysis techniques extending the HAP system. LIRA is the acronym of *Low-Count Image Restoration and Analysis*. The software was developed by Alanna Connors, David Esch, David van Dyk and Nathan Stein, who initially described the analysis in two papers (see Esch et al. 2004, Connors and van Dyk 2007). So far it has not been used in conjunction with H.E.S.S. data. The following chapters try to explore the capabilities of the LIRA software package. It will become clear that LIRA indeed can overcome some limitations of HAP, two of the biggest advantages compared to HAP are that LIRA can deal with low-count images in a mathematically correct way, and eradicate the PSF distortion. Thus it can in fact provide a more detailed view of low-count sources. Both, advantages and disadvantages are systematically analysed in the following.

LIRAs main algorithm uses a Bayesian statistical method for deconvolution of low-count images X, where the value of pixel i can be described as $X_i \overset{d}{\sim} \text{Poisson}(\Lambda_i)$.[1] For an ideal observation process, Λ_i would be the true astronomical source distribution (the spatial structure of relative intensities). In reality however, observations are distorted for various reasons. Firstly, there is background noise b. Secondly, the instrumental response distorts the true source image. This is described by the point spread function $\text{PSF}_{i,k}$ which defines the probability that a photon which should be projected into pixel k is recorded in pixel i. And lastly, the observation process is not equally sensitive over the whole image area. Reasons for this are for example differences in observation time spent on different regions in the sky or differences in the detection probability for gamma ray events from different zenith angles. The combination of these effects is here summed up as the exposure ϵ.

[1] The notation is similar to the one in Esch et al. (2004):‘A random variable X is said to follow a Poisson distribution with parameter or intensity Λ if $Pr(X = x) = e^{-\Lambda}\Lambda^x/x!$. In this case $E(X) = \Lambda$ and we often write $X \overset{d}{\sim} \text{Poisson}(\Lambda)$ (read as X is distributed as Poisson with intensity Λ). This representation conditions on the intensity parameter, Λ, which in turn may vary.’

With these definitions, the expectation value for the observed counts in the image can now be described as:

$$\chi_i = \sum_k \text{PSF}_{i,k}(\Lambda_k + b_k) * \epsilon_k \tag{10.1}$$

and the observation becomes: $X_{i\sim}^{d}\text{Poisson}(\chi_i)$. [1]
Of interest for the analysis of observations is the inverse of this problem: calculating $p(X|\Lambda)$, the probability of observing X given the true source distribution Λ. This probability is identical to the likelihood function $L(\Lambda|X)$. But although LIRA includes an algorithm[2] to calculate a maximum likelihood estimator (MLE, the Λ which maximises the likelihood function) the main algorithm of LIRA tries to get more than the best fit. It allows also for an estimation of the uncertainties of the restoration via its Markov chain Monte Carlo (MCMC) algorithm.[3]

Bayesian Formulation of the Deconvolution Problem

Finding the true source based on Equation 10.1 is called an ill-posed problem. Which means that unlike in well-posed problems, there exists no unique solution.
To be able to infer probability information about Λ given X, it is necessary to define a model of the *joint probability distribution* of the *prior distribution* $p(\Lambda)$ (also called *unconditional* or *marginal distribution*) and the *data distribution* $p(X|\Lambda)$ (see Gelman et al. 2003), which can be written as:

$$p(\Lambda, X) = p(\Lambda)p(X|\Lambda) \tag{10.2}$$

And following from that, conditioning on the known value of X is the *posterior distribution*:

$$p(\Lambda|X) = \frac{p(\Lambda, X)}{p(X)} = \frac{p(\Lambda)p(X|\Lambda)}{p(X)} \propto p(\Lambda)p(X|\Lambda) \tag{10.3}$$

10.1 The Prior and Hyperprior Distributions

Prior distributions capture the knowledge of the possible values or shapes of the true source Λ, regardless of the observation (or *prior* to observation, hence the name). If, for example, it would be known, that the observed object has an elliptical shape, this knowledge could be put into the prior distribution as a parametric model.

[1] This assumes that the acceptance for γ-rays and for background events is equal. This is in general not the case. LIRA is in its current implementation not able to include different exposure maps depending on the nature of the event (γ-ray or hadronic in origin). This is a field where LIRA definitely has to be improved in order to be fully accepted for H.E.S.S. data analysis. This is a shortcoming of LIRA when applied to H.E.S.S. observations. It can not be solved in the current implementation of LIRA, so this has to be left for further work on the LIRA code base to implement different exposure maps for background and source. For the further analysis, the acceptances are therefore assumed to be equal.

[2] The expectation maximization (EM) algorithm

[3] If not noted otherwise, the term 'LIRA' is used instead of 'LIRA run with the MCMC algorithm' for brevity reasons from now on.

LIRA does not follow this approach of pre-selecting certain parametrized source shapes, but instead uses a more general model, the NK model, which has been already described in Section 9.5.2. This model represents an image by its split proportions on multiple scales. LIRA assumes, that these split proportions follow a distribution *per scale*, but does not assume that these distributions are known a priori. Therefore the split proportions are fit along with Λ itself. The prior distributions of the split probabilities D (see Equation 9.17) can be described by parametric distributions like the Dirichlet distribution, which is used in LIRA:

$$D_{k\sim}^{d} Dirichlet(\alpha_k, \alpha_k, \alpha_k, \alpha_k) \tag{10.4}$$

One consequence of choosing a common prior distribution per scale k is that the parameters $A = \alpha_k, k = 1, ..., K$, can be seen as smoothing parameters. The smoothing parameters determine how regularized (smoothed) the posterior distribution will be, but the best value of A is not known in advance. Therefore, again, a prior distribution is chosen for A and the values of A will not be fixed but fitted together with the split proportions D. Since this new prior distribution is the prior distribution of the prior distribution parameters, it is called *hyperprior distribution.*

In LIRA, the *hyperprior* was chosen to be

$$p(\alpha_k) = \alpha^{\kappa_1} * e^{\kappa_2 * \alpha^{\kappa_3}} \tag{10.5}$$

with two parameters fixed: $\kappa_1 = 0$, $\kappa_3 = 3$ and one parameter set to the default $\kappa_2 = 1000$ (can be changed per simulation). As explained in Esch et al. (2004), the decision for choosing this type of hyperprior and the default parameters was based on empirical tests. In Section 10.6, example images will be shown for different values of κ_{1-3} together with problems that could be connected with this choice of hyperprior distribution.

Another prior distribution is used for governing the total amount of counts in the multiscale model G (see Equation 9.17) which is assumed to follow a Gamma distribution: $G_{\sim}^{d}\text{gamma}(\gamma_0, \gamma_1)$ (for more details on the reasons for choosing these distributions, see Esch et al. 2004). This second prior distribution is not very critical, for it is tuned to have on average the same amount of counts in the reconstructed image as in the original observation. This is a reasonable and uncontroversial assumption.

Markov Chains and the Gibbs Sampler

Markov chain Monte Carlo methods are a way to simulate distributions by drawing samples of approximate distributions and then correcting these draws to better approximate the prior distribution (see Gelman et al. 2003). A Markov chain is a sequence of random variables Λ_i, that have the Markov property, which means that given any $i = I$, the distribution of Λ_i with $i > I$ does not depend on the values of Λ_i with $i < I$, but only on Λ_i. While the first property of MCMC systems, the improvement along the chain, makes it a useful tool for fitting a target distribution, the Markov property allows for short sampling iterations, because the draws can be considered independent after a few steps. Independent draws are needed to have an unbiased estimate of the prior distribution.

Markov chain simulations are often used when calculating the prior distribution directly is either (computationally) impossible or inefficient. Instead, a Markov process is constructed then, which draws simulated values $\Lambda_i, t = 1, 2, 3, ...$, from a transition distribution $T_i(\Lambda_i|\Lambda_{i-1})$, that depends on the previous draw, and which improves over time such that after a certain period, the transition distribution converges to a stationary distribution equal to the prior distribution $p(\Lambda|X)$ (Gelman et al. 2003). The simulation is therefore run long enough for the process to get close enough to the stationary distribution such that the samples of the chain can be assumed to come from the true prior distribution.

This description shows that it is critical for any MCMC simulation to check that convergence is reached, maybe even in several different chains, depending on the constructed chain. The length until this state is reached is called the *burn in* length. It is also necessary to ensure that there is no in-sequence correlation, which is done by rejecting all but every n-th draw, a process called *thinning*.

While the checks themselves are critical, their presentation in this text has been omitted in most of the following test-source analyses. In the case of the main object of interest, the simulation of HESS J1837-069 in Chapter 11, all the necessary steps are shown in detail, including the burn in and thinning checks.

Several algorithms are used to construct Markov chains, with the most famous being the Metropolis-Hastings algorithm and the Gibbs sampler. LIRA uses a Gibbs sampler to construct the chain. This sampler is a special kind of Markov chain algorithm, which is often used for multidimensional problems: the correction of the draws after each step is only done for a subset Θ_t of the parameters $\Theta = (\Theta_1, \ldots, \Theta_d,)$, while all other parameters are kept fixed. Normally, one iteration is defined as d of these steps, after which all parameters have been updated.

Cycle Spinning

One visual artefact that Haar wavelet decompositions (with decimation of wavelet coefficients for de-noising) and also representations through the Nowak and Kolaczyk model have in common is a block like structure. This structure is created in the decomposition process, because some regions of the image are closer to the discontinuities of the decomposition basis. The intensity of the artefacts is thereby dependent on the exact position of the original feature in the image. Features close to the center of the image for example are, in the NK model, close to all split edges, while other features which are fully in one part of the highest level split don't show the same size of artefacts. One method to avoid these artefacts is called *cycle spinning* and was introduced by Coifman and Donoho (1995). The basic idea is that the input image can be shifted and the resulting reconstructions can be combined so that artefacts average out. This procedure is for example applied in the TIPSH (Translation Invariant Poisson Smoothing using Haar wavelets) algorithm,

which, as the name says, applies this procedure to wavelet transformations using the Haar wavelet.
Cycle spinning is also applied in LIRA, where the translation is randomly chosen at every cycle.

Null-Model

Unlike the description in the first paper about the LIRA analysis (Esch et al. 2004) suggests, the software at its current stage is not only fitting a multiscale model to the data, but it can now at the same time scale a null-model or background model. The combined model is then $\mu = f * \mathrm{M} + \mathrm{MS}(\alpha)$, where f is the background scale factor, M the null-model and $\mathrm{MS}(\alpha)$ the multiscale model.
When interpreting the LIRA results, the null-model can mean one of two things.
Firstly it can be seen as the true background, which is often a flat image of an arbitrary value, in the best case close to the true value. Since LIRA includes the exposure map in its calculation, a flat image which corresponds to a uniform noise from all directions, is a good approximation in most cases. If the null-model is assumed to contain the noise, then the reconstructed multiscale representation contains all the source counts. This is the goal of many wavelet based algorithms, removing noise (insignificant wavelet coefficients) on all resolutions.
But secondly, the null-model can additionally contain structures, e.g. point sources or extended structures of known shape, which are known to exist in the FoV of the image. If this is the case, the resulting multiresolution image can be thought of as the source-subtracted image. Unlike a mathematical subtraction however, the structures in the null-model are modeled during the fitting process to take the Poisson nature of low count observations fully into account.
What is to note here is that LIRA is not able to fit the position of a feature of known shape but unknown position. Errors in the null-model have to be assessed from the background scale parameter or have to be detected in the resulting image. The question to ask is therefore not 'How does the emission region look like without a known source shape (e.g. point source)?', but something more like the following: 'Given a known source shape (e.g. point source) at this certain position, what is it's likely intensity and how would the residual emission look like?'.
To answer this question with LIRA, several additional steps are required. These will be shown in Section 10.6.10 for the Galactic center region. Before we come to this point, some more information on LIRA is required.

10.2 A Detailed Look at the LIRA Output

The LIRA software is written in the statistical language R and in C. The input files can either be supplied in the R friendly ASCII-table format, or alternatively, a basic FITS import wrapper can be used.

LIRA provides a set of raw output data, which require further treatment before any interpretation is possible. Apart from the images which are produced either during the sampling of the posterior distribution or during the iterative steps of the expectation maximization, there are a number of parameters which are constantly logged. These include the iteration step, the log-likelihood of the current posterior distribution, the smoothing parameters of the different scales, the number of counts which were placed in the multiscale component by the fitting procedure (instead of being placed in the null-model), and the scale of the null-model.
The following paragraphs explain, what these individual parameters can tell about the analysis, while the information contained in the output images is explained in the next sections.

Scale Parameter of the Null-Model

The scale of the null-model (or background) is an indication of how well the background and the intensity and position of additional source structures are known. If the shape and the intensity of the background are well known, then the scale parameter should be close to one. Since LIRA contains two main algorithms (EM and MCMC), the scale parameter can be accessed as either the last log entry in the case of expectation maximization or the mean of the logged values in the MCMC mode (after the burn in phase and with thinning of the remaining samples)[1]. The second method also allows to access other characteristics like e.g. the variance of the scale factor.

Number of Counts in the Multiscale Model

If the fitting procedure was used to infer whether no additional multiscale components are present in an image (apart from the specified null model), it is not enough the check whether or not the scale parameter is close to 1. The multiscale component must also not contain a significant amount of counts as well. As in the case of the scale parameter, the counts in the multiscale component are derived from either the mean value of the samples in the case of a Markov chain, or the last value in the case of the expectation maximization algorithm.
Checking whether the amount is significant or not, cannot be determined by a single LIRA run, but requires three steps. Firstly, a best fit multiscale component has to be determined and the number of counts in the multiscale component have to be checked. Secondly, observations of the best fit, without the multiscale component, have to be simulated while care must be taken to include the PSF and the exposure. Thirdly, the same fitting procedure has to be applied to the simulated observations. Then, the amount of counts in the multiscale component of the fit for the fake observation has to be compared

[1] Accessing parameters or images based on the MCMC samples always needs to include burn in and thinning to ensure that the samples are independent. If they can not be assumed independent, then e.g. the mean of a parameter might be biased.

to the amount in the multiscale component of the real observation. If both are compatible, there is no evidence for a multiscale component.
These steps are described in detail for fake test sources to explore and confirm the procedures and capabilities of LIRA. Later, this procedure is applied to the source region of HESS J1837-069.

Smoothing Parameters and Log-likelihood

The log-likelihood of the posterior distribution and the smoothing parameters of the different scales are needed for additional checks. A converging of the MCMC chain for example should result in a distribution of a specific smoothing parameter around a fixed value. If that smoothing parameter shows several modes instead, a simple mean of the images and output parameters can not be used. Since the images of the individual fitting steps are logged as well, the logged values of the smoothing parameters are not used for reconstruction purposes.

10.3 Pixel-wise Significance

One of the advantages of LIRA over simpler wavelet-based deconvolution and fitting techniques is the availability of the full posterior distribution. The LIRA output images (the draws Λ_i from the stationary distribution of the Markov chain, which contains the multiscale model of the source distribution without the scaled null model) not only allow to infer a best fit image, which is created from the mean of the output images, but enable the inspection of the output images with other statistical methods. Interesting quantities are e.g. the variance and the significance. Under the assumption that the pixel values are normally distributed, the mean divided by the variance gives a measure of the statistical significance of the pixel values.
Important about these significance maps is one specific fact: the pixel values of adjacent pixels are not independent. This has consequences for the usefulness of these significances as maps, because normally, we are not interested so much in the significance of an individual pixel, but in the significance of a source or emission region. Since adjacent pixels are correlated, the significances in all pixels of a source (assuming we can define these pixels) cannot be summed or otherwise simply combined, to get a source significance. These problems are similar to the ones of the HAP significance maps, where the significance values are also correlated and therefore do not in general allow to infer on source or sub-source significance. What has to be done instead, to gain information about the sources significance, is to make use of LIRAs feature to scale a background or null-model and sample the residual distribution.

10.4 Feature-wise Significance

The last section dealt with pixel-wise significance, which corresponds to the probability that the counts in a pixel are above the background and therefore originate from a non-background source. This pixel-wise significance can be extended to a detection significance, if the counts of all positions which are assumed to belong to a source are summed up and compared to a corresponding background region. A similar procedure is done in the correlated significance maps in HAP. In HAP, a circle is used to collect pixel values, but this method is not useful to actually distinguish between sub-features which have sizes of the same order or smaller than the size of the circle. And even if it were, one problem persists with this method: the significance is assessed under the assumption that photons can either come from the background or the sub-feature region. If other sources or sub-features are nearby (within the reach of the PSF), their possible contribution to the counts has to be taken into account as well. A significance that takes the contribution of nearby sources into account, but still combines all photons of the supposed sub-feature region will be called *feature-wise significance* in the following.

One of the advantages of LIRA in combination with other general analysis techniques is the ability to access this feature-wise significance without resorting to oversampling or re-binning. The drawback of this method is, that a shape for the feature has to be known in advance.

The posterior mean of LIRA draws gives an estimate of the deconvolved true source distribution. This is essential to be able to define a feature which is of the same size or smaller than the PSF. In the case of HESS J1837-069, the southern part of the emission can be seen as such a feature. Its size is in the same order as the PSF and the deconvolved image helps to clearly separate the main emission from the southern emission. It can be seen in Figure 9.1 that the HAP excess does not allow a clear separation because of the overlap of the wings of both features, the main hot-spot and the southern part.

Unfortunately, calculating the feature-wise significance with LIRA is not straight forward, but includes a number of steps.

10.4.1 A Vision Model

To get these information from the analysis of the region, it is necessary to identify the different emission regions. An automatic procedure which extracts the different disconnected source regions and applies additional analysis steps is called a *vision model* (Starck and Murtagh 2006). Typical vision models for astronomy are the SExtractor (source extractor, see Bertin and Arnouts 1996), or the MVM (Multiscale Vision Model, see Bijaoui, A. et al. 1998).

However, none of these models uses the LIRA analysis. Thus, a suitable vision model for this analysis had to be developed. It relies heavily on the LIRA deconvolution and consists of the following steps, which for simplicity reasons here also include the feature-significance steps, which are not actually part of the vision model:

1. Image reconstruction:
   A LIRA analysis with the pure background shape as null-model is conducted. The input image is the ‘on’ map from the HAP analysis. The main LIRA results are a

reconstructed image (the mean of the LIRA draws excluding the burn in and with thinning applied) and the background scale factor.

2. Feature definition:
   A general best way to define a sub-feature in a source region can not be given. It depends on the purpose of the analysis and might include prior information taken from multi-wavelength observations or theoretical models of the feature. Such a more complex algorithm was used to model the point sources in the example application on data from the Galactic Center in Section 10.6.10. There, the position and shape (point like) of two sources were assumed to be known. The 'masking' in that case required a fit of a sharp 2D-Gaussian such that the remaining counts at the source position matched the counts in the surrounding pixels: the fit has to satisfy the assumption that the extended emission around the point source has no special property at the location of the point source.

   However, to be able to infer knowledge about the existence of distinguished emission regions without prior information, a blind feature search is necessary. For this vision model, the search consists of two steps: de-blending and labelling.

   For the deblending algorithm a *white top-hat transform* is applied to the LIRA reconstruction of the image. This step is necessary, because in the LIRA reconstruction, which is intrinsically a smoothing operation, the wings of neighbour object profiles can blend and thereby lead to a merging of objects.

   The white top-hat transform is defined as the difference between the image X and its opening by a structuring element b: $T_{white}(X) = X - X \circ b$, where $\circ$ is the opening operation. The opening operation itself is defined as the dilation of the erosion of an image X by a structuring element b: $X \circ b = (X \ominus b) \oplus b,$ where $\ominus$ is the erosion operation $(X \ominus b)(i) = \sup_{h \in \mathbb{R}}(X(i-h) + b(h))$ and $\oplus$ the dilation operation $(X \oplus b)(i) = \inf_{h \in \mathbb{R}}(X(i+h) - b(h))$ (see Heijmans and Ronse 1990).

   Top-hat transformations (white and black) are typical operations in mathematical morphology, and are often used for feature extraction. They 'clean' the image of small fluctuations and filaments but also remove features larger than the structuring element.[1]

   The decision between determining the boundaries of features clearly on the one hand and including as many counts from the original picture as possible on the other hand has to be made at some point. The decision for this analysis was done based on the idea that if an increase in the size of the structuring element[2] does not increase the number of counts in the feature significantly (a factor of 0.1 was chosen here), the

[1] For future work on this topic, the search for a more elaborate technique might be a good starting point. For example does the described filter technique ignore all multi-scale information which is internally created in LIRA. LIRA does currently not export this information, but implementing an export routine and a feature separation based on the higher scale information might be an interesting extension to the current LIRA analysis. It will mainly improve the sensitivity of the feature separation for large extended features.

[2] The structuring element is a circle approximation in a (2n+1)x(2n+1) pixel sized square, where n is called the size in this thesis.

feature is assumed to be captured. This additional check is performed to ensure that the size of the structuring element does not artificially shrinks the feature region.
The next step is to apply a labelling procedure to the resulting image. This labelling simply searches for connected pixels in the image. Since the term 'connected' requires a definition of 'containing something' a threshold is used to make this distinction. This threshold is needed, since the reconstructed image contains the mean value of each pixel over all simulated values and therefore even pixels that can not be considered part of the source can have a small value, but still a value different from zero. The threshold level for this analysis was chosen such that 90% of the full counts are in regions assumed source features, the remaining 10% are assumed background. For the previously shown test sources as well as the analysis of HESS J1837-069, this corresponds to about 70% of the number of pixels. The labelled array of HESS J1837-069 can be seen in Figure 11.6.

3. Source null-model simulation:
   The next step is to run a LIRA analysis again, with a simulated observation from the null-model. The null-hypothesis is that a previously defined feature is just a fluctuation of the background. Therefore an null-model is created by using the reconstructed image from step 1, but with the corresponding labeled regions set to zero, corresponding to pure background. From this image, an observation is simulated taking the background (also from step 1), the exposure, and the psf into account. This second LIRA analysis can be called reference-analysis, since it provides the reference to compare the original analysis to.

4. Feature-wise significance determination:
   The probability that the previously detected feature is just a fluctuation of the background and the surrounding sources is directly given by the probability that the observation from the null-hypothesis and the real observation lead to comparable result in the LIRA analysis.
   The comparison is made based on a summary statistic, which tries to capture the full, multi-dimensional differences in the posterior distributions: for the further analyses in this thesis, the number of counts in the multiscale model was chosen as this summary statistic.[1] The probability that original observation and simulated observations are from the same source distribution is then given as the overlap between both probability distributions. This overlap was calculated by fitting a PDF of a normal (or exponential) distribution to both statistics and calculating the overlap p = $\int_{-\text{inf}}^{\text{inf}} min(PDF_1(x), PDF_2(x))dx$. This overlap has sometimes been called OVL in the literature (Inman and Bradley 1989, Reiser and Faraggi 1999).

[1] It must be noted that although this has the advantage that it is widely unchanged by the modifications made to LIRAs hyper-prior distribution, the actual proof that this captures the difference in the distributions must be left for future work on this topic.

10.5 Source Shape Significance

Information about the shape of a source is hard to acquire. Often, one can see significance maps created with HAP, which have a color scale that shows a switch from blue to red at some value. The effect is comparable to a cut at this significance level. The remaining (red) regions in the image are seen as the shape of the source – while not claimed anywhere, this is nevertheless the effect of such a map presentation on the viewer. The information contained in the significance map however does not actually describe the source shape, but the most likely position of a single source under the assumption that the source is shaped like a circle. One can not determine the certainty/significance of the shape of a source based on this information.

From the samples of the posterior distribution in the LIRA output, this certainty can be accessed. This information could be called source shape significance. The idea is that the shape of the reconstructed source has a major influence on the summary statistic. For weak features, this will be the number of counts in the multi-scale model or, since it is anti-correlated to this parameter, the background scale.

LIRA places counts in either the background model or the multi-scale model, which leads to the disappearance of weak features or the wings of features in the multi-scale model, if more counts are assumed to be part of the background. It is therefore possible to sort the draws of the posterior distribution by the background scale factor and create images of the draws for levels of certainty in that parameter. Choosing a threshold level on the counts-per-pixel, a contour of the source shape can be drawn for e.g. the lower 5%, the main 90%, and the upper 5% of the LIRA draws. These contours can then be seen as significance levels on the shape of the source (but not as indicators for how significant the emission is compared to the background).

The source shape significance does not directly help in the quest to identify sub-features. It merely gives a confidence on the deconvolution procedure, that means whether the attribution of counts to either the background, or certain parts of the image is stable / justified by the data. If different possible source shapes could result in the same observation, the source shape significance shows this fact as a spread in the contours.

10.6 Testing LIRAs Capabilities With Simulated Sources

While the theoretical foundations of LIRA have been published in several papers, there are at least two open questions with respect to applying LIRA to H.E.S.S. data: Since some of the input parameters and predefined functions are chosen based on empirical tests conducted by the authors and are not chosen for theoretical reasons, the usefulness of these defaults need to be tested for H.E.S.S.-like data. And since the actual implementation of the software has not yet received widespread adoption, the actual implementation must be tested as well. Such tests will be described in this section.

The features of LIRA described in the two main papers dedicated to this software seem impressive and are exactly what could help the morphological analysis of HESS J1837-

069. It nevertheless seemed necessary to test those features in controlled setups to see if and in which detail the results of the software could be trusted.

Relative Error as a Measure of Reconstruction Quality

Sometimes, the relative error of two images with I pixels each, defined as

$$RE = \sqrt{\frac{\sum_{i=0}^{I}(N_{reconstr.,i} - N_{source,i})^2}{\sum_{i=0}^{I}(N_{source,i}^2)}} \tag{10.6}$$

is used as a measure of reconstruction quality. However, the usefulness of this number in morphological analysis is doubtful.

The reason can be seen in the following example: Given a true source in the shape of an E, and two different reconstructions A and B (see Figure 10.1) with the same RE value, reconstruction A captures the morphology of the true source clearly better than reconstruction B.

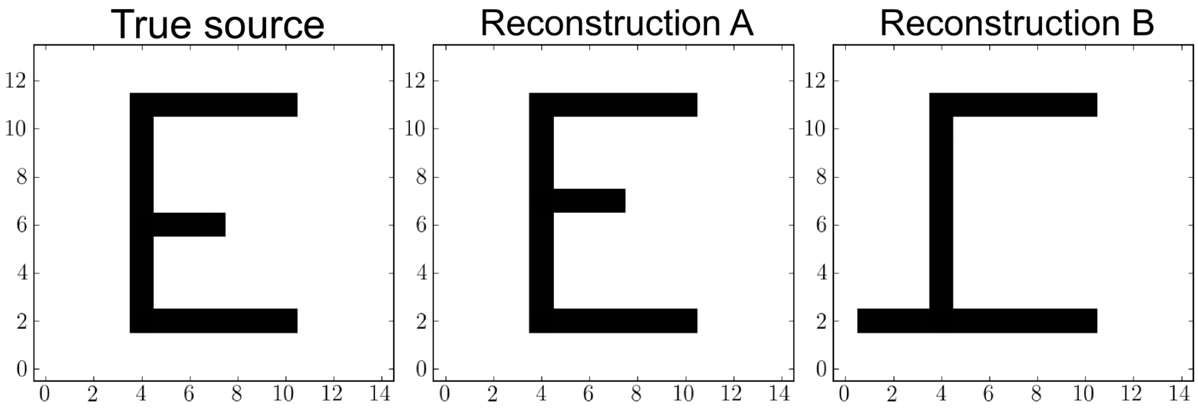


Figure 10.1: Example of a true source morphology and two different reconstructions with the same RE.

The reason for this discrepancy between the RE and the perceived reconstruction quality is that the RE is not localized. The same number of counts in reconstruction B are further away from the true source than the same number of counts in reconstruction A. This localization is not taken into account in the RE. However, defining this localized error (as a single number) seems hard, because the shapes of sources cannot generally be described by simple parametrised models.

It is necessary to compare the true source shapes and the reconstructed source shapes with the help of residual images by eye.

In the following, tests will be described which have been conducted to learn the limits of the restorations made by LIRA. The tested features are the quality of the deconvolution in shape and intensity as well as the determination of the null-model scale factor for different kinds of sources, namely pure backgrounds, point sources, and extended sources (each with absent as well as present background in varying intensities).

To summarize the LIRA procedure again, here are the steps to test the LIRA reconstruction: The tests start with a true source distribution, a psf, a true background and an instrument response, summarised in the exposure map. An observation is constructed by convolving the true source and the background with the psf and multiply the sum by the exposure. Then a random sample is taken from this map, resulting in the simulated observation.

This observation, the psf, the exposure map and the shape of the background (but not the scale) are given as input to LIRA. Among the output of LIRA is then the reconstructed source (which can be compared to the true source) as well as the background scale, (which can be compared to the true background divided by the background shape given as LIRA input).

10.6.1 Flat Background

LIRA is able to fit a shaped background simultaneously with the multiscale component. The shaped background is thereby scaled by a scalar value. LIRA is not able (for now) to use a parametric model which could be adjusted to shift or tilt the background or even change its shape. If the background shape is known from other simulations or if the region of interest is small enough so that a flat background can be assumed, LIRAs capabilities should be interesting enough, though, since fitting the background and the source at the same time is more than the standard HAP analysis can do at the moment.

The first simple test is therefore to see, whether LIRA is able to pick up the scale of a flat low-count background in the absence of any other component. At the same time, a HAP style background determination is applied to the observations, in order to provide a visual comparison between the LIRA reconstruction and the standard reconstruction.[1]

[1] This is not how background determination is done in HAP, however, the basic principle of using 'off' regions to determine the background is kept. From the 'off' regions – pure background regions – HAP calculates a background value normalized to the same area which is used for the 'on' counts (1 pixel in the case of the tests). Here, however, this procedure can be simplified: no sources means that the 'on' and the 'off' region are identical. The 'on' regions are pixels, so the mean of several 'off' regions can be taken as the background value, but the 'off' region has to be multiplied by the exposure before that to provide an exposure corrected background value.

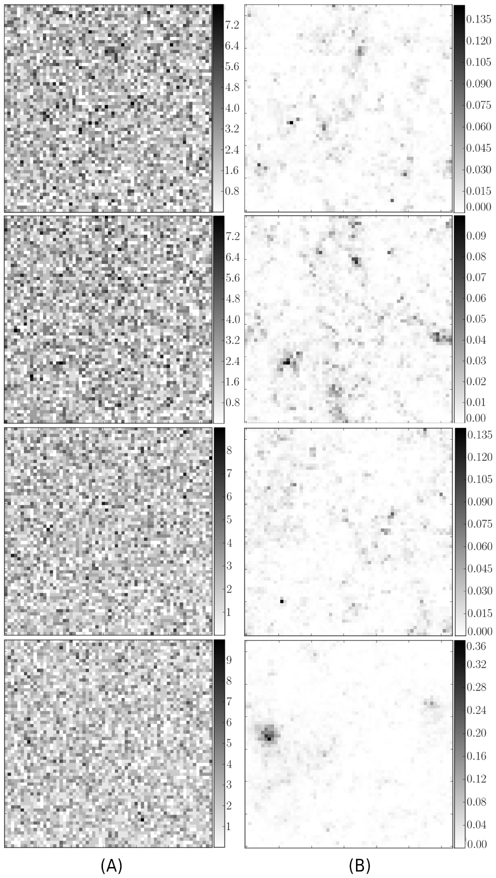


Figure 10.2: *(A)* The simulated observation of a flat background with a nonflat exposure map. *(B)* The posterior mean of the multiscale component that was fit at the same time as the background. The 'hot spot' in the fourth image has a maximum value which is a sixth of the background value and sums to about one count, so it can not be considered a real 'hot spot'. A significance check has however not been conducted here.

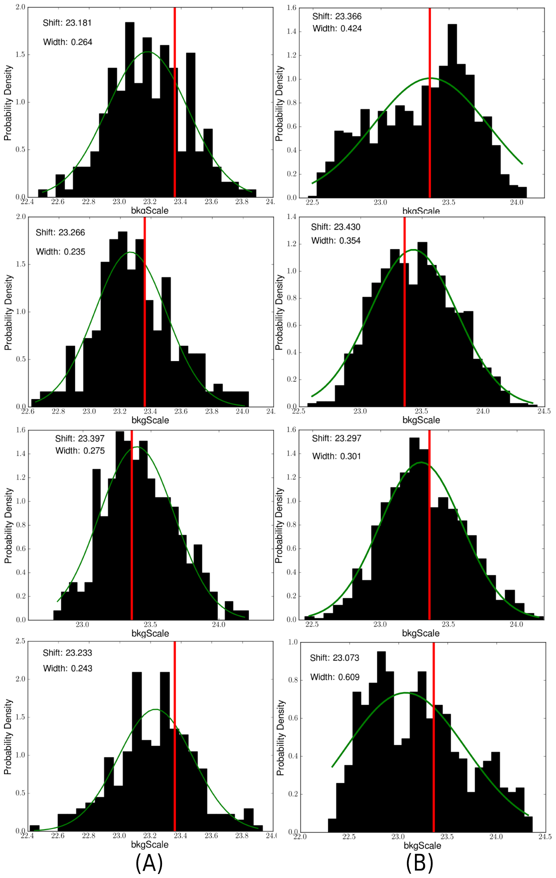


Figure 10.3: *(A)* Distribution of the backgroundscale as simulated by LIRA. The red line shows the true value. *(B)* in comparison, the backgroundscale as it would be calculated by using the mean of a 32x32 pixel 'off' region. All possible sub-images where used and the distribution of the background scales is shown here.

To test this, the observation of a 64x64 pixel wide image was simulated. The procedure was as follows: a flat image with 2.335 counts per pixel was multiplied by a shaped exposure map (which was normed to its maximum value). The resulting pixel values were then taken as the expectation value of a Poisson distribution. Four random draws of that distribution were taken, and on each an independent LIRA chain was created with a flat background model of value 0.1 counts/pixel (which means that the reconstruction of the background scale is expected to be 23.35) and the true exposure map. The results can be seen in Figure 10.2 and 10.3. From the posterior mean images, it can be seen, that the emission of the multiscale component is very low.

In initial tests, the reconstructed scale was off by 0.7 counts per pixel, which is 50% of the

true value! This was due to a bug in the LIRA software which scaled the background by the *given* exposure map, but the multiscale component by the *normed* exposure map.
If both components use the normed exposure map, the reconstruction is very good with values between 2.318 and 2.339 counts/pixel. What becomes apparent in the comparison between the methods is the difference in the spread of the reconstructed values. While the Markov chain samples the distribution around the true value rather sharply (with the fitted normal width between 0.024 and 0.028), the standard HAP method has a fitted normal width between 0.030 and 0.060, which means that the standard method is more sensitive to fluctuations in the sample regions. While LIRA is slightly better in low-count background determination, it requires a priori knowledge of the background shape, which the standard HAP methods do not. There is however no reason not to combine the two methods, using the ring-, template- or other background maker from the HAP analysis to define the shape of the background for different observation conditions and then use the background shape as the null-model in a LIRA analysis. For the analysis in this thesis however, a flat background shape was assumed.

10.6.2 Point Sources

The second test includes a PSF that was taken from the HAP analysis of HESS J1837-069. A 64x64 pixel wide image of two point sources was created. Each point source has an intensity of 200 counts (see Figure 10.4), the image was convolved with a PSF and from this convolved image, a simulated observation was taken (see Figure 10.5).

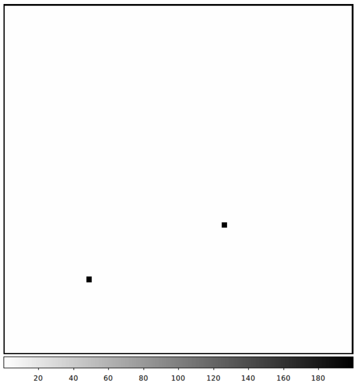


Figure 10.4: A simulated 'true source' region consisting of two point sources.

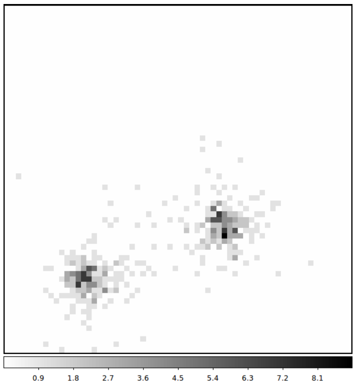


Figure 10.5: A simulated observation of the fake source with a PSF comparable to the one from the HAP analysis of HESS J1837-069.

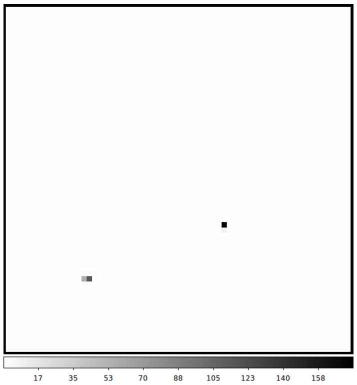


Figure 10.6: The reconstructed source distribution from 2435 LIRA draws . The two sources can be seen clearly, their position and intensity are well reproduced.

The exposure map was in this example assumed to be flat with value 1. No background was used for this test (later tests will have a background, but this is explicitly stated in

each test). Those simulated observations and maps were then used as input for a LIRA analysis. Five independent LIRA runs in Markov chain Monte Carlo mode, with 10000 draws each with a burn in of 250 draws and a thinning of 20 were conducted, resulting in 2435 draws. A flat background was fitted to the image. From these 2435 remaining draws, the mean of each pixel value was calculated. The result can be seen in Figure 10.6. The reconstructed image clearly shows the two point sources. The positions of both sources are perfectly reconstructed if the pixels with the maximum values are taken. If the positions are calculated as weighted means, an offset of about 0.02 - 0.3 pixels can be seen (first source position in x-y coordinates is: (13, 15) and the reconstructed position is (13.02, 14.69). The second source position is (23, 40) and the reconstructed position (22.95, 39.96).
The intensities in the true source image were 200 counts for each point source. This was very well reconstructed in one source with 201.84 counts, and closely with 190.32 counts in the second source. To determine the number of counts in the reconstructed image, they were summed up in a ±5 pixel region around the source positions, which is the same region used for the center of mass calculation.

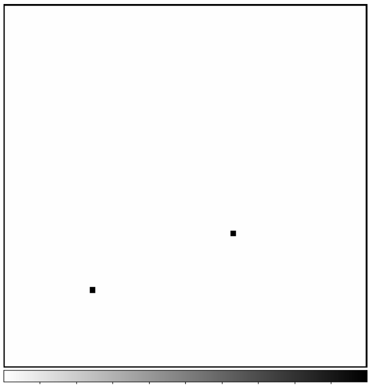


Figure 10.7: Second LIRA test with two point sources: A fake source with two point sources.

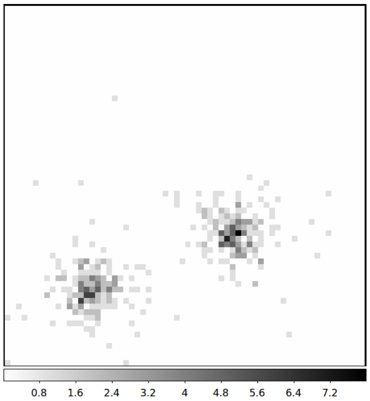


Figure 10.8: Simulated observation of the fake source with a PSF comparable to the one from the HAP analysis of HESS J1837-069.

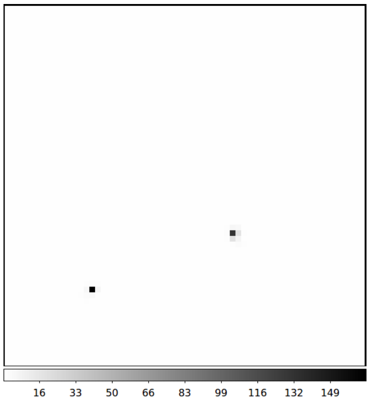


Figure 10.9: Reconstructed source distribution from 2437 LIRA draws.

The same analysis was done a second time, but with a different 'observation', to rule out that the first observation was simply a too good fit to the source. The results can be seen in Figure 10.5.
Determined in the same way as in the first test, the intensities of the sources are now 191.52 counts and 199.17 counts, compared to the 200 counts in the true source. The positions of the point sources are: originally (13, 15); reconstructed (12.99, 14.97) and originally (23,40); reconstructed (22.88, 40.17). The results differ a bit compared to the first test, as expected, but they are comparable. The intensities and positions of the sources

are in good agreement with the true source.

The result of these tests is that the LIRA software is well suited to reconstruct point sources, both in position and in intensity. With regard to the intensity a summation of the pixel values around the position of the point source is necessary. This test should of course be repeated with different positions and source intensities, however, since for HESS J1837-069 point sources are not the main interest, the tests with point sources were stopped here.

10.6.3 Extended Sources

During these tests with the LIRA software, shortcomings in the standard random number generator in R became apparent. To increase the number of draws of an analysis, several Markov chains were started at the same time which ran in parallel. The result however showed identical 'random' draws in the chains and therefore the chains lead to identical results. As it turns out, this is known to the R developers in principle: the R manual states that 'Initially, there is no seed; a new one is created from the current time when one is required. Hence, different sessions started at (sufficiently) different times will give different simulation results, by default' (The R Development Core Team 2011, library/base/html/Random.html). How different the times have to be, however is not documented.

To avoid this behaviour, a 'SNOW' (Simple Network Of Workstations)[1] wrapper was written for LIRA as part of this thesis, that now includes MPI (Message Passing Interface) support and can run several chains in parallel. SNOW includes a random number generator, which can be used for parallel R sessions.

Since the main focus of the morphological analysis of HESS J1827-069 is under the assumption that the true source distribution might be an extended source, simple point source tests are not enough to show what LIRA is capable of. Therefore tests with extended sources were carried out. The sources were created with a procedure that starts at a random position in the image, then does a random walk on the image and deposits less and less (random) counts in the underlying pixel at every step. Several of these random walkers where started on each image. This procedure generates arbitrarily shaped sources. To simulate the observation procedure, a constant background was added and the resulting image was folded with a PSF. Random values where produced then from the pixel values, assuming they are the expectation values of a Poisson distribution. The exposure was assumed to be constant, just like in the tests with the two point sources.

[1] http://www.sfu.ca/ sblay/R/snow.html

Extended Source With Standard Parameters

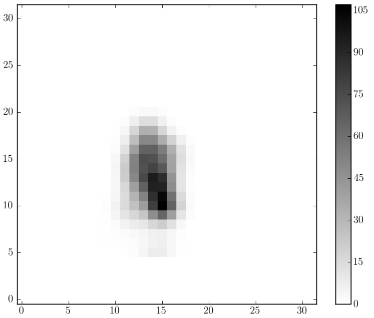


Figure 10.10: LIRA test with extended sources: A fake extended source.

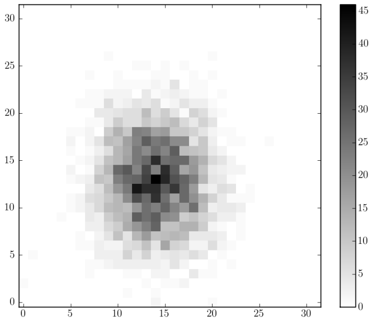


Figure 10.11: Simulated observation of the fake source with a PSF.

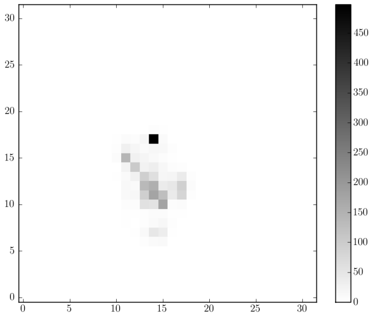


Figure 10.12: Reconstructed source distribution from 1.500.000 LIRA draws (burn in: 250, thinning: 30). LIRA was run with standard parameters.

A first test with an extended source was carried out (still without a background). The true source can be seen in Figure 10.10 and the simulated observation in Figure 10.11. The LIRA reconstruction in Figure 10.12 is the mean of 1.500.000 LIRA draws, after the draw sequence was thinned by 30 and a burn in of 250 was chopped off.
When comparing the reconstruction with the true source, large differences become apparent: the reconstructed source seems to have a hot spot in the north and a fainter extended source in the south. The true source does not show these features.
While the reconstruction does not match the true source morphology, it is a **possible** reconstruction based on the information which is preserved in the observation and the PSF. This can best be seen when the true source and the reconstructed source are both convolved with the PSF and compared to each other (see Figures 10.13 and 10.14).

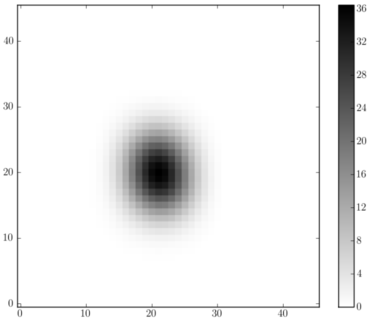


Figure 10.13: The fake extended source convolved with the PSF.

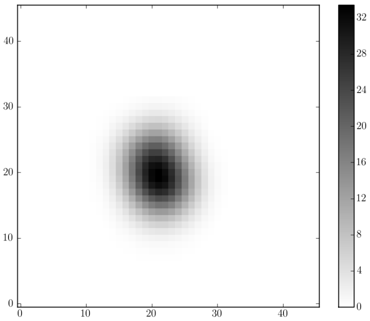


Figure 10.14: Reconstructed image convolved with the PSF.

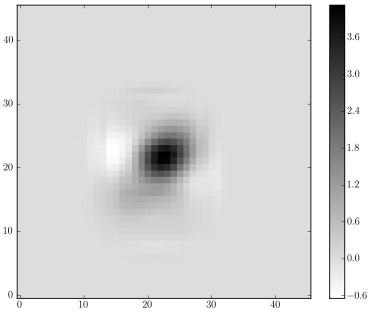


Figure 10.15: The difference between both images. It can be seen, that the differences are small compared to the values of the true source.

Both images are very similar in shape and even the values are very well reconstructed: The difference image in Figure 10.15 shows that the mismatch is in the order of 3% of the values of the true source.

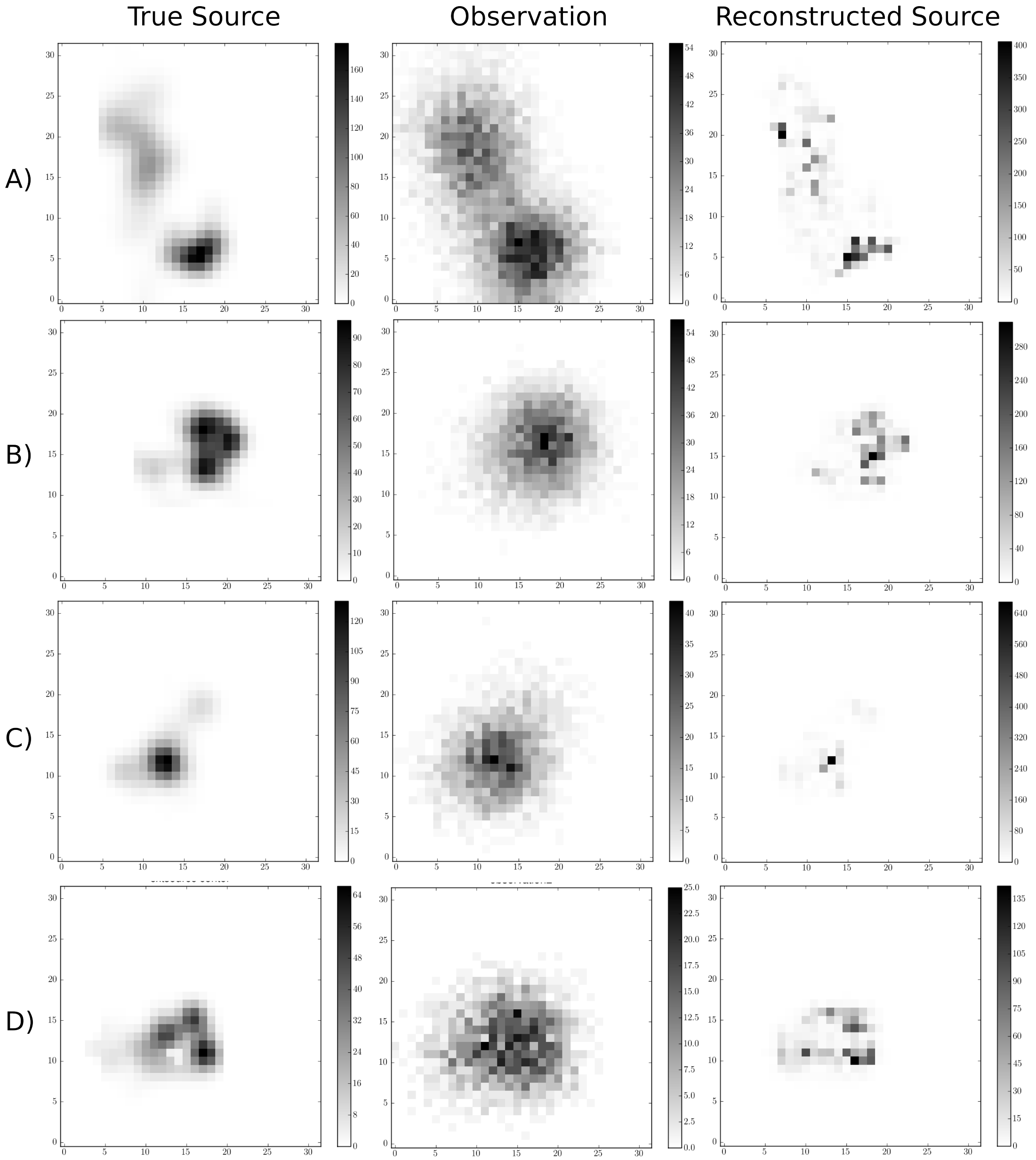


Figure 10.16: Several simulated test sources analysed by LIRA with default parameters of the hyperprior distribution. The results all look like beeing composed of point sources, this might be caused by the hyperprior that LIRA uses. The fact that LIRA is also able to reconstruct the images much closer to the true sources can be seen in Figure 10.18

But this does not mean, that the reconstruction worked like expected. The posterior mean should be a good approximation of the true source, or at least should the posterior distribution contain the true source as a possible mode. This mismatch between extended sources and reconstructions can be seen in several different tests. Figure 10.16 shows a number of different extended sources.

10.6.4 The Influence of the Hyperprior

The mismatch between the true extended sources and the posterior means makes the LIRA software not suitable for image reconstruction of extended sources, at least for sources similar to the ones created for the tests described above. This is however only true for the default functions implemented in LIRA: A possible reason for this mismatch could be the hyperprior (see Section 10.1), which was chosen based on empirical reasons by the authors of LIRA. The explanation for the currently implemented hyperprior was, that a hyperprior function which forces the prior parameters to zero, would allow the observation to influence the outcome of the reconstruction to a bigger extend (Esch et al. 2004).

Forcing the parameters of the Dirichlet prior distribution to zero leads to big differences in the adjacent pixel values. The posterior means which can be seen in Figure 10.16 could be described as consisting of point sources wherever a point source or an extended source would lead to the same observations. Note that selecting point sources over extended sources is equivalent to selecting smaller Dirichlet parameters over large ones. Therefore, it seems as if the choice of the hyperprior and its default parameters could lead to these results.

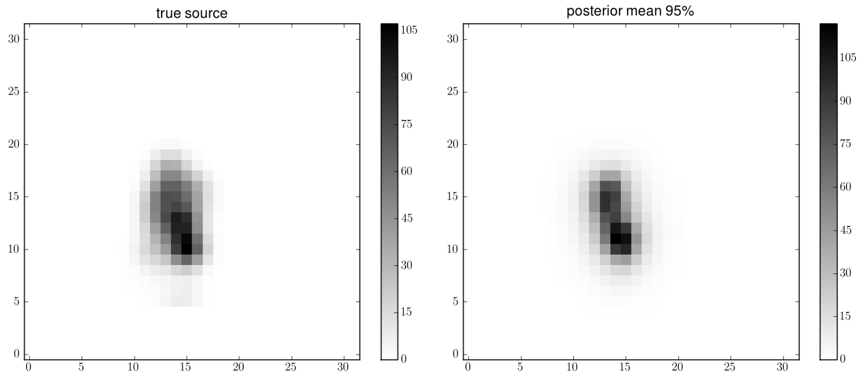


Figure 10.17: The true source *(a)* and the reconstructed image *(b)* are very similar. This posterior mean was created from 5 independent LIRA runs with the smoothing parameter 0 set to 1, instead of letting LIRA fit that parameter via the hyperprior. The mean was taken from 9900 samples, after a burn in of 100 was cut and a thinning of 5 was applied to each chain.

This of course needs to be tested. If the result of the LIRA runs would be more similar to the true source when run with a hyperprior not forcing small values, the software could be adapted to be a very useful tool in the analysis of H.E.S.S. observations. Implementing a different hyperprior in the Gibbs sampler is not trivial. Therefore, a different method was chosen which involved less code changes: the smoothing parameter for the lowest level at the pixel resolution was (instead of being changed by the hyperprior) set to be 1.

By setting this parameter value to a value higher than 1, the image would be forced to be smooth on the pixel level. Being 1 means, that the observation can still influence the outcome in one or the other direction. The other alpha parameters are still fitted according to the default hyperprior and Gibbs sampler.
The result of this change in the software can be seen in Figure 10.17. The reconstructed image is now much closer to the true source, although the observation (see Figure 10.11), the exposure, and the PSF were unchanged. The shape as well as the individual pixel values are in very good agreement with the true source.

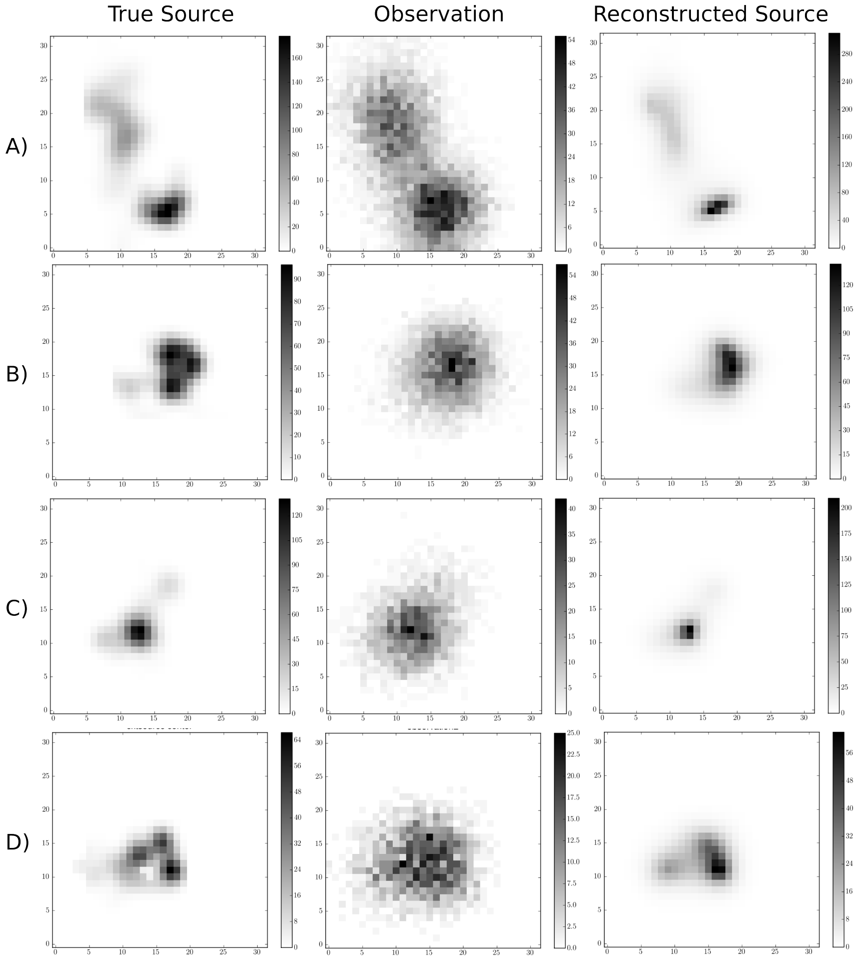


Figure 10.18: The same simulated extended sources as in Figure 10.16, but with a modified hyperprior. The reconstructed image is the posterior mean of five independent chains of a modified LIRA, with 10000 iterations each, a burn in of 250, and a thinning of 30. The modification to the standard LIRA was a fixed smoothing parameter at the pixel level with a value of 1.

The same simulated extended sources which were shown in Figure 10.16, have been analysed with the modified LIRA software, where the smoothing parameter 0 was set to 1. The result of this analysis can be seen in Figure 10.18. The 'reconstructed' images are the posterior mean of five independent LIRA chains, each with 10000 iterations, a burn in of 250 and a (very generous) thinning of 30. This means that the mean is taken from 1625 uncorrelated samples. When comparing the two reconstructions of the extended sources, it is clear, that the smoothing parameter which controls the split probability at the pixel level really has a big impact on the reconstruction. At least for extended sources, it is necessary to allow the prior parameters of the Dirichlet function to also take values above 1.

10.6.5 Weak Sources

To see the full power of LIRA, it is not enough to run the tests above. As the name 'Low count Image Restoration Analysis' implies, the important part of LIRA is the statistically correct handling of observations with low counts. In the analysis of HESS J1837-069 in Section 11, the number of counts per pixel is in fact well below 10 for the biggest part of the image (see Figure 9.7). It is therefore necessary, to see how well the algorithm works with observations containing counts per pixel in this range. If the underlying distribution was Gaussian, as it is often assumed in other analysis methods, there wouldn't be the need for additional test observations. The previous observations could simply be scaled to lower values and the results of the analyses would be the same. But in the regime of pixel values below 10, new simulated observations have to be created, containing again only integer values.
Because of these reasons, further tests were necessary. To be able to compare the lower count with the previous higher count restorations, the same true source shapes as well as identical other parameters were used (still without a background).
The results of this test can be seen in Figure 10.19. The observations which can be seen in that figure are taken with a flat exposure of 0.1. This means that the reconstructed sources should be in the order of 0.1 of the true source intensity and therefore much weaker than in the previous tests.
In source *A)*, the intensity and position of the southern main spot are well reconstructed, but also the weaker, 'S' shaped part of source is well rebuild. Obviously not in every detail, but it shows the right length and the bridge between the main source and the other part of the source can be seen. What is missing are the weak wings of the main part of the source and the weak wings of the 'S' shaped part of the source.
Source *B)* is only roughly reconstructed. The three stronger spots of the source can not be seen, but are replaced by a single spot, which is slightly stronger than the three spots of the true source. The weak wing on the west side which can be seen in the true source is not reconstructed. What can be seen in this area instead, is a slight extension of the main spot. Still, the overall reconstruction is good enough to identify morphological differences compared to circular or elliptical shapes.

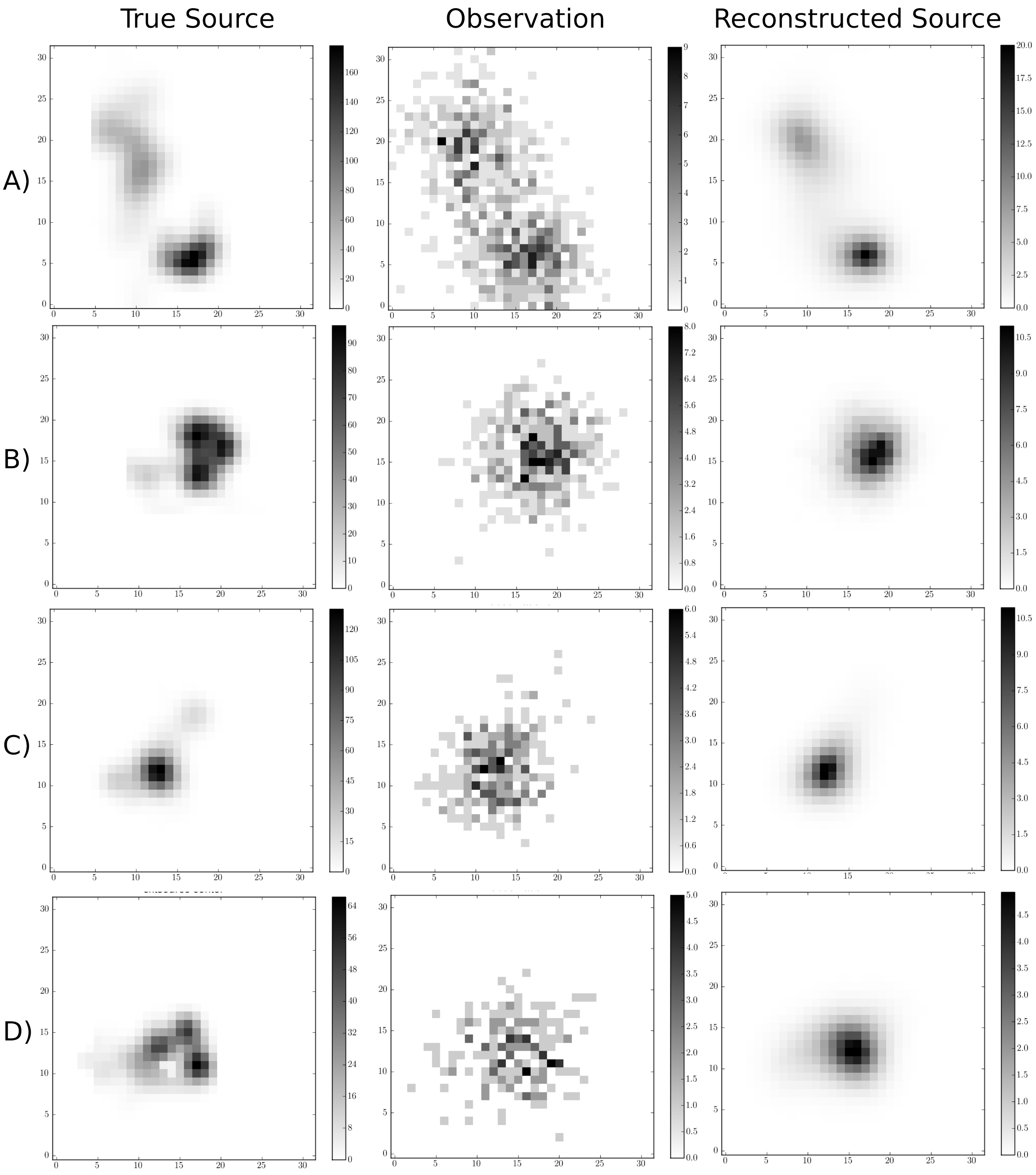


Figure 10.19: The same simulated extended source shapes as in Figure 10.16 and Figure 10.18. But here, the observation was taken from an flat exposure of 0.1 which means a weaker source. The reconstructed images are the posterior mean of five independent chains of a modified LIRA, with 10000 iterations each, a burn in of 250, and a thinning of 30. Again, the modification to the standard LIRA was a confinement of the smoothing parameter at the pixel level to 1.

In source *C)*, again, position and intensity of the main spot are well reconstructed. The northern extension of the source can be seen, but it is much weaker than in the true source. The same is true for the smaller southern extension. Overall, the faint features of the source can not be seen, but this might be in part a problem of this specific observation. No other, 'better', observations were created, because doing so would add a bias in the overall review of the capabilities of LIRA.

Source *D)* consists of three spots grouped around a 'hole'. The hole is not reconstructed. The only features which can be seen in the reconstruction are the triangular shape and the weak western extension. The intensity is well reconstructed, but since the three spots are not reconstructed individually but modeled as a single broader spot, the peak intensity is lower then in the true source.

Despite some limitations in regard to details of the source shapes, the reconstructed sources in these tests come very close to their originals. Whats strikes positively compared to other deconvolution techniques is the absolute absence of reconstruction artefacts in all tests. Note also, that the intensities of the sources are very well reconstructed.

10.6.6 Weak Sources Plus Background

What has been tested in the last two tests is the ability of LIRA to reconstruct images in the sense of modeling a posterior distribution under the assumption of a given PSF and exposure. As already mentioned, this can be described as deconvolution – but successful deconvolution is not the only and not the most interesting feature of LIRA.

The next feature which needs to be tested in order to fully make use of LIRA, and for the ability to infer feature significance, is LIRAs ability to scale a null-model while at the same time sampling the posterior distribution.

In the following tests, simulated weak sources combined with a flat background were analysed. Both the source and the background are in the same order of magnitude (between 0 and 10 counts per pixel) and therefore both follow a Poisson distribution which could only badly be approximated by a Gaussian.

In these tests as well, the reconstructed images seem quite good (see Figure 10.20) – in general as good as the reconstructions in Figure 10.19. Therefore, the details of the shape reconstructions are not explained again.

As a general result, however, it can be said that the inclusion of a background with counts per pixel in the same order of magnitude as the source counts does not change the reconstruction in a notable way. This again is a big advantage of the LIRA analysis, for H.E.S.S. data analysis has to deal with background of comparable intensity to the source intensities. Therefore the background subtraction is a challenge not every analysis program can deal with equally well. The following goes into greater detail in regard to the background problem.

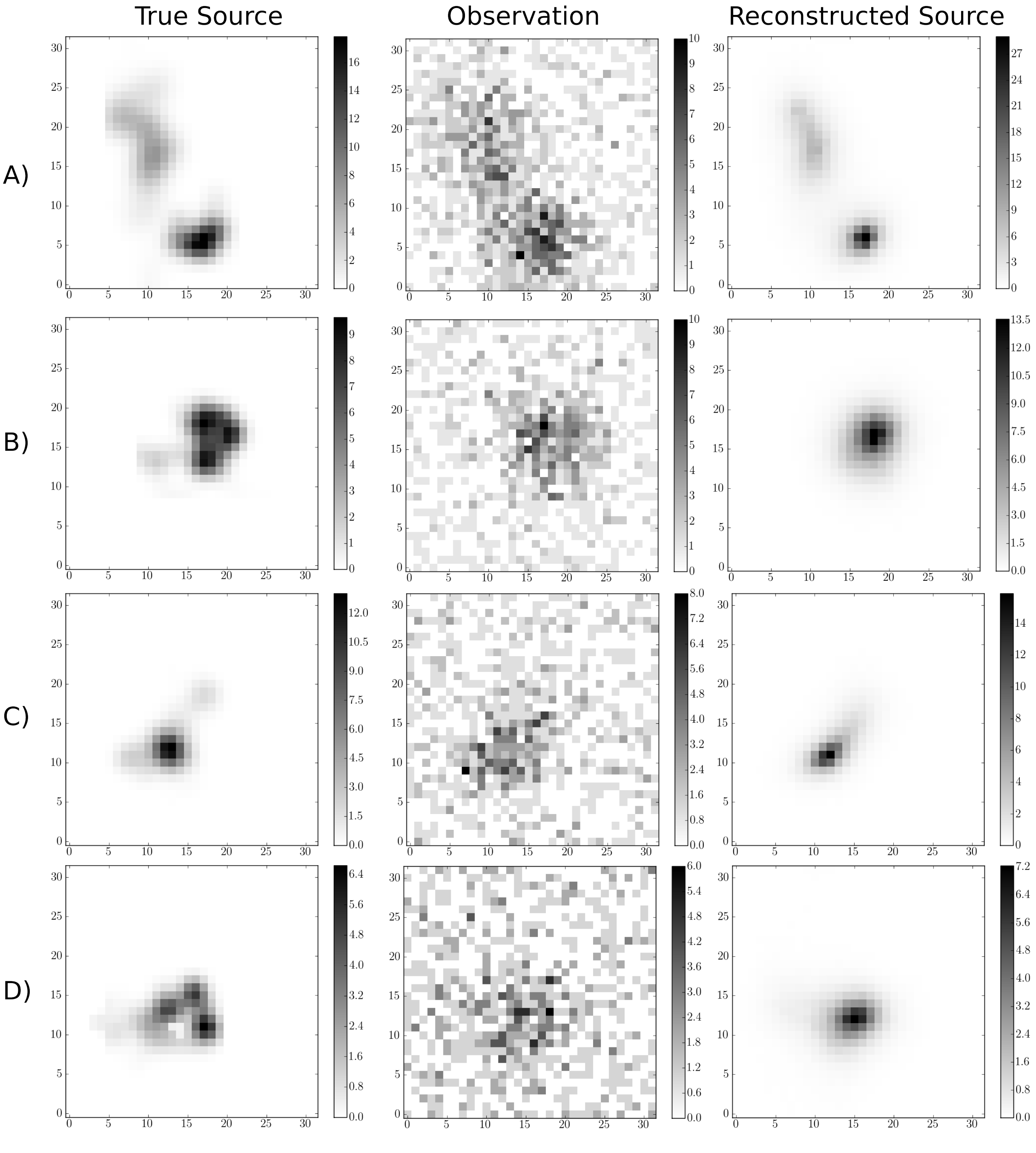


Figure 10.20: The same simulated extended sources as in Figure 10.16, 10.18 and 10.19. The observation was again taken from an flat exposure of value 0.1, but a flat background of value 0.5 counts/pixel was added. Each reconstructed image is the posterior mean of five independent chains of a modified LIRA, with 10000 iterations each, a burn in of 250, and a thinning of 30. Again, the modification to the standard LIRA was a confinement of the smoothing parameter at the pixel level to 1.

10.6.7 The Influence of the Background Intensity

One important question for the reconstruction of weak sources is the question of background determination. Does the software pick up the right value and how does the background reconstruction influence the intensity and shape of the reconstructed source?
In the 'classical' background subtraction techniques, a halo around the source region can suggest, that parts of the source were used to calculate the background scale. This means the background regions were not determined correctly. Apart from the region around the main source, local fluctuations (positive and negative ones) in the residual image must be considered natural artefacts of the subtraction process. To get rid of these artefacts which can distract from the real sources, a common technique is to show these residual images (so called excess maps) only in a 'smoothed' version. Most often, smoothing means convolving the residual image with a Gaussian of some size, for example the size of the PSF. The PSF size is justified by the assumption that fluctuations smaller than the PSF can only be produced by chance. And since convolving with a Gaussian of the size of the PSF in a background subtracted image will level all smaller structures, the remaining parts are considered real. It must be noted, that this technique is normally not used for quantitative morphological analysis, but only for overview pictures to guide the scientist for future investigations. Nevertheless, images produced this way, when seen as a guide to the source region, have a big influence on which further analyses of the source region are conducted, where slices are produced, and which initial values for 2d Gaussian fits are chosen.
In this 'classical' method, the background is determined by observing regions in the sky which are supposed to contain only background. Such runs are called 'off' runs. Alternatively, without dedicated observations, background regions in the image can be selected. In the case of separate background observations, great care must be taken to keep all other parameters (like zenith-angle and exposure) comparable. In the case of in-image background determination, the exposure correction is important, because background-only regions are often at the edge of the image.
With LIRA, the situation is a bit different. LIRA does never determine a background shape, like the HAP background makers do, but it relies on the user to provide the right background shape. There are two possible ways of dealing with this. Either, a background is given and kept fixed during the Markov chain, or a background shape is given, but the scale is modified together with the other parameters. In the second case, no 'off' observation is used and no part of the image is chosen by the user to contain pure background. [1] The question is therefore, if LIRA picks up the right background level in the presence of a source, and, more importantly, if the size of the image region containing pure back-

[1] The background in all analyses in this thesis was assumed to be uniform and only the scale was adapted by LIRA. This is a shortcoming of the presented analyses, which can however be eased by the fact that only hard cuts where used for the creation of the HAP maps, which removes about a factor 7 more background events than the standard cuts (Funk 2005). Since LIRA uses only a single exposure map for null-model and multiscale model, a systematic error is introduced in the background level determination. This systematic error must be studied further, but this is beyond the scope of this thesis.

ground has an influence on the reconstruction quality. It might be possible that LIRA has bigger problems with determining the background level when there is only a small area of background in the image. It might also be easier for the program to identify the background level when the image contains larger parts without sources. The following test-runs therefore compare LIRAs abilities when confronted with different background region sizes to give a feeling for LIRAs abilities in that regard. They are however not systematic studies, which would require many more simulations.

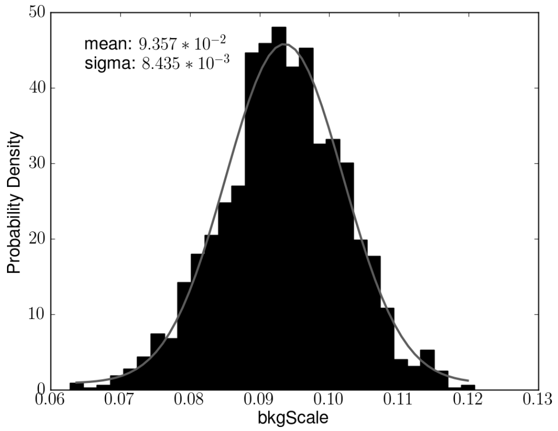


Figure 10.21: Test 1.1: Distribution of the background scale parameter. The true value is 0.1. The reconstruction was done with the extended source A) (see Figure 10.19). The image is 32x32 pixels wide.

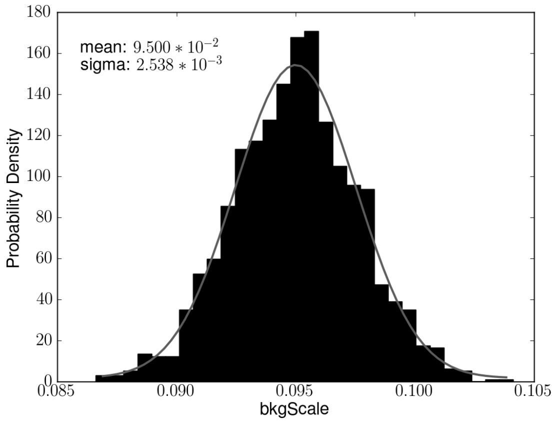


Figure 10.22: Test 1.2: Here, the source image of test 1.1 was embedded a large area of background. The image is therefore 64x64 pixels wide.

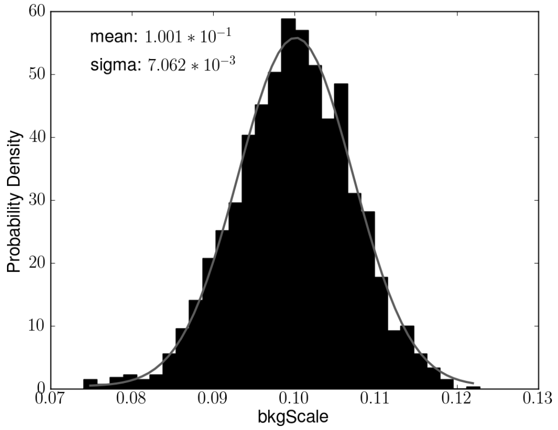


Figure 10.23: Test 2.1: Distribution of the background scale parameter. The true value is 0.1. The reconstruction was done with the extended source B) from Figure 10.19. The image with the source is only 32x32 pixel wide, it hardly contains any pure background pixels.

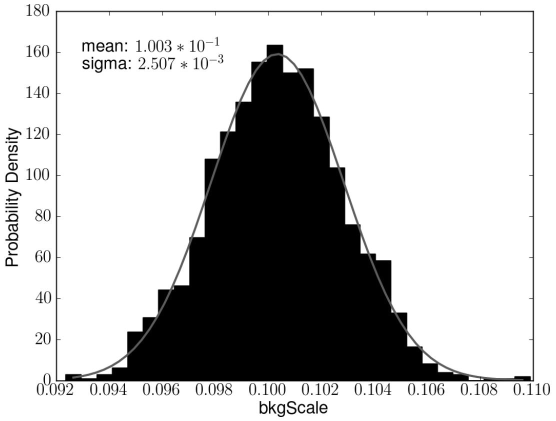


Figure 10.24: Test 2.2: Here, the source image of test 1.1 was embedded a large area of background. The image is therefore 64x64 pixels wide.

In the following two tests (Tests 1.1/1.2 and 2.1/2.2), again a flat background was fed into the LIRA analysis and the background scale was fitted (the true source can be seen in Figure 10.19). The intensity of the background was 0.5 counts/pixel (the true background scale parameter however is 0.1) as in the tests before. The only difference between the two runs of each test is the size of the images: In each second run (Test 1.2 and 2.2.) the images from the first runs were padded with pure background and thus their size is twice as big as in the first runs (Tests 1.1 and 2.1). The result is that the second runs are done with images containing the same source image but surrounded by a larger frame of background. The smaller images have a size of 32x32 pixels and the bigger a size of 64x64 pixels. The results can be seen in Figures 10.21 and 10.22 as well as Figures 10.23 and 10.24.

Contrary to the initial concerns, LIRA does not have problems picking up the background. All four test results look very similar to each other. Test 1 shows, that the true background scale of 0.1 was very well reproduced in both runs: the smaller image (with 295 pixels of pure background compared to 729 with source contributions) has a background scale of 0.094 ± 0.008 and the bigger image (with 3221 pixels of pure background compared to 729 with source contributions) of 0.095 ± 0.003. While the bigger image produces an only marginally better mean value, the spread is smaller and therefore reconstructed with more certainty – which was expected.

Also in the second comparison it can be seen that the background scale was reproduced very good: 0.100 ± 0.007 in the smaller image and 0.100 ± 0.002 in the bigger image. Again, the spread of the distribution is smaller in the bigger image, but the mean value of the distribution shows no significant difference.

The results is very promising, because it seems unnecessary to have large parts of the image containing pure background to be able to reconstruct the background scale. It is however to be expected that this strongly depends on the strength of the source. Systematic tests, which would require many more simulations, have not been conducted for this thesis. However, the ratio of two third of the pixels containing source contributions and one third containing pure background seems to be a good test for the source HESS J1837-069.

From these tests it can be concluded that LIRA is capable of fitting the true value of a low count background, even in the presence of weak extended sources. It also gives a quantitative measure of the error on the best fit value. Unlike other methods, it does not require the explicit selection of source-free background regions.

The only drawback compared to the HAP background makers, though, is the requirement to have a known background shape. If the background is non-flat, its shape must be measured independently from LIRA, but can then of course be used.

10.6.8 The Influence of the PSF

In some of the previous tests, it became obvious, that the LIRA reconstruction was not able to reveal the finer details of extended sources.

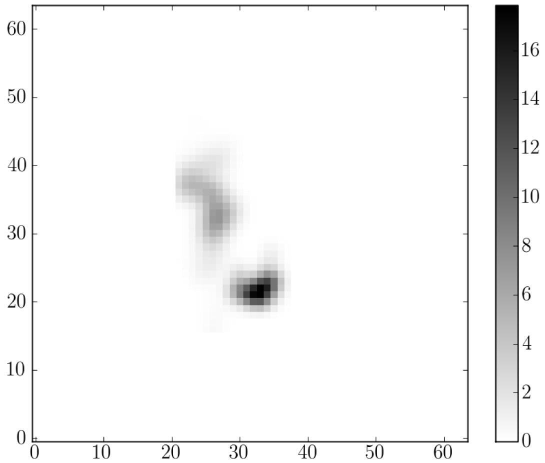


Figure 10.25: *(a)*: The true source morphology.

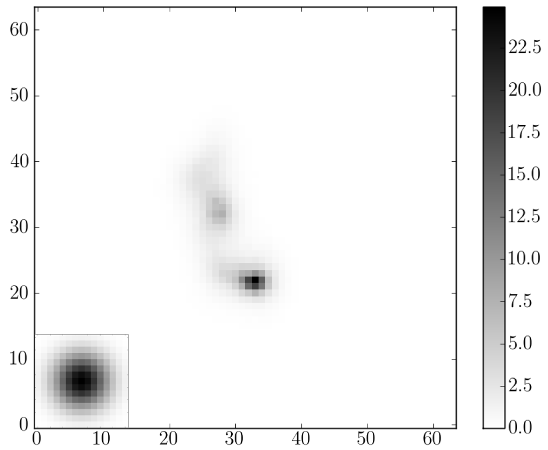


Figure 10.26: *(b)*: Reconstruction from an observation with a wider PSF. The PSF is shown in the lower left as in the following figures.

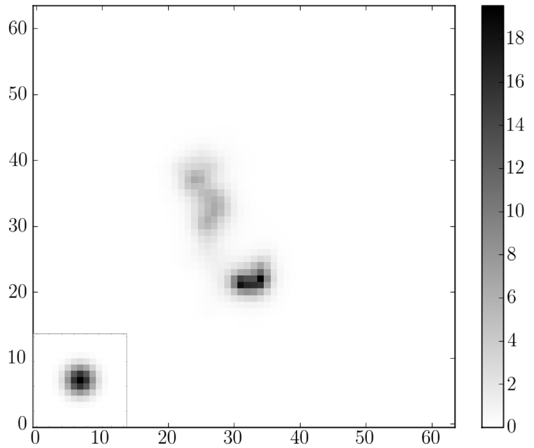


Figure 10.27: *(c)*: Reconstruction from an observation with a sharper PSF.

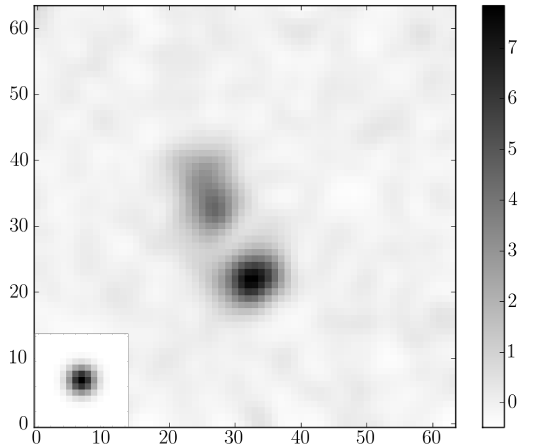


Figure 10.28: *(d)*: HAP like excess map created by subtracting the background from an observation with the sharper psf.

It is clear that having longer observation times can overcome these limitations and reveal finer details. Similarly, substructures in weak sources become visible when the PSF is sharper. To see the influence of the sharpness or size of the PSF on the reconstruction of weak sources' details, tests with a sharper PSF were created (involving the sources from the previous tests).

Some results of these tests can be seen in Figures 10.25 to 10.28. Image *(a)* shows the true source. Images *(b)* and *(c)* show reconstruction with a wider and a sharper PSF, whereas image *(d)* shows a HAP like excess map, created by subtracting the best-fit background from the observation.

Image *(b)* is the reconstruction which was created for Section 10.6.7, with the PSF shown in the lower left corner for comparison. Image *(c)* is created from the same true source *(a)*, but with a sharper PSF, as can also be seen in the lower left corner. Although image *(b)* is already a good reconstruction of the true source, the second reconstruction is much better, revealing finer details like the 'banana' shape of the main spot and a more pronounced 'S' shape of the extended feature. The result that a finer PSF leads to finer details was expected. In both versions features can be seen which are of the same extension or smaller than the PSF, for example the 'S' shaped extended feature in image *(b)* shows features which are smaller than the PSF and which are present in the true source. Very compelling is, after all, the improvement which this LIRA result offers over techniques of background subtraction and Gaussian kernel smoothing. This becomes obvious when comparing image *(c)* to image *(d)*.

10.6.9 Summary of Results of Tests with Simulated Sources

All tests shown so far used simulated source, not real sources and tested them under different conditions. The main results so far are:

LIRA is a tool well suitable to reconstruct point sources in position and shape (point sources are reconstructed as point sources).

The modified LIRA (with the smoothing parameter at pixel level fixed to 1) is even very good at reconstructing extended sources.

The reconstruction of the background level works fine, even if the true source covers large parts of the image and not many pure background pixels are available.

The case of choosing a wrong background model / null model is shown in the next chapter. In general, LIRA is not able to determine if a model is wrong, because it will simply scale the background accordingly and compensate this by placing more counts in the multi-scale model. Therefore, the simplest way to check for a correct background assumption is to provide LIRA with a background model which has the best-known shape and the best-known intensity. Then, the LIRA scale factor should be close to 1, or otherwise the best-known background is not the best-fit.

The past tests were used to establish trust in the image reconstruction capabilities of LIRA for known sources which capture the main characteristics of HESS J1837-069, like extension, intensity, etc. The further question about feature separation and significance will be discussed later in Chapter 10.4.1. However, this feature separation and significance relies on a good image reconstruction, therefore these tests were necessary.

Test of the Vision Model on Known Sources

The vision model from Section 10.4.1 can be tested in a basic way by applying it to the previously shown test sources. It can be visually shown that the features defined by the vision model align well with the original source shapes. The shape of the original source, shown as contour lines, is overlaid over the shape defined by the vision model.

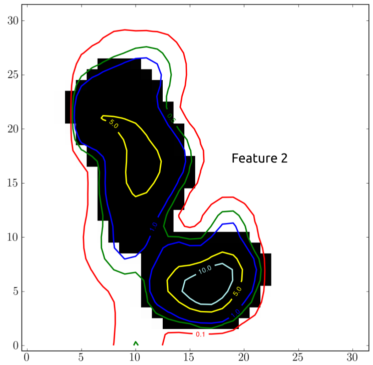


Figure 10.29: Comparison of contours of the true source with the shape of the feature detected by the vision model. The number of counts in the true source which are outside the feature boundaries are 40 of 782 ($\sim 5\%$). For comparison, the number of background counts in the same region (covered by the true source but not the detected feature) are 329.

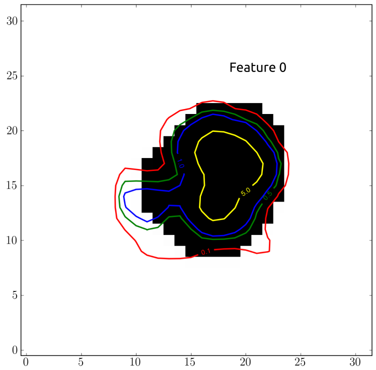


Figure 10.30: Comparison of contours of the true source with the shape of the feature detected by the vision model. The number of counts in the true source which are outside feature boundaries are 15 of 396 ($\sim 3\%$). For comparison, the number of background counts in the same region are 150.

It has to be noted that the feature selection in the first example in Figure 10.29 was on the edge of detecting two separate features. The vision model requires less than 1% difference between the number of counts in features from successive structuring element sizes in order to accept the features, but the difference in size 2 to 3 of Feature 0 was 2% and therefore close to being accepted. The features defined by this size of the structuring element can be seen in Figure 10.34 together with the number of counts in the features with respect to the size of the structuring element in Figure 10.33.

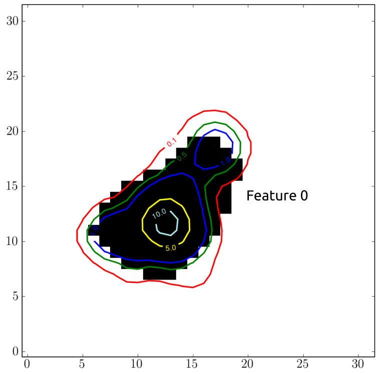


Figure 10.31: Comparison of contours of the true source with the shape of the feature detected by the vision model. The number of counts in the true source which are outside feature boundaries are 18 of 267 (∼ 7%). For comparison, the number of background counts in the same region are 338.

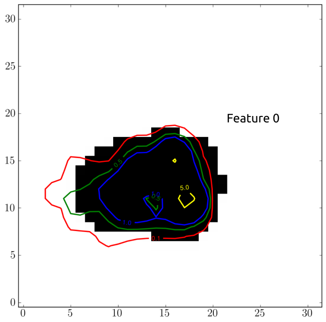


Figure 10.32: Comparison of contours of the true source with the shape of the feature detected by the vision model. The number of counts in the true source which are outside feature boundaries are 8 of 222 (∼ 4%). For comparison, the number of background counts in the same region are 68.

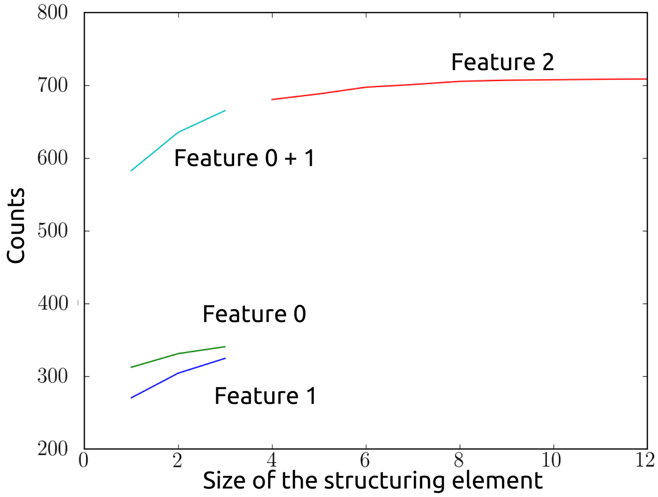


Figure 10.33: Number of counts per feature versus different sizes of the structuring element. Feature 0 is close to being detected as a separate feature, but the initially defined threshold of 1% increase is not reached.

The full series of features dependent on the size of the structuring element can be seen in Figure 10.34. Of course, the influence of the labeling threshold on the separated features can be shown as well, which is done in Figure 10.35. Varying the structure size however leads to a smoother increase in the feature size, which results in a simpler stopping criterion.

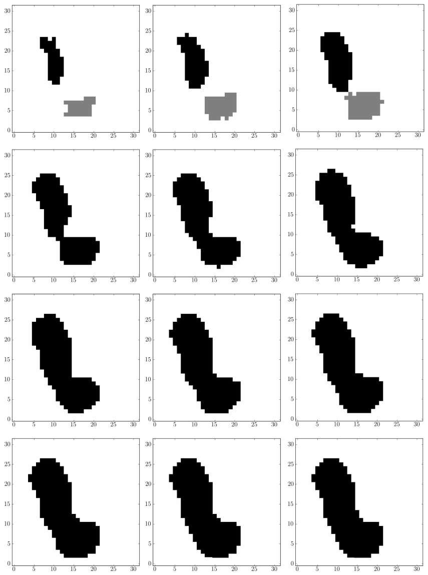


Figure 10.34: Feature separations based on different white top-hat structure sizes. The final feature separation is chosen such that the number of counts in the feature does not increase by more than 1% when going to the next structure size.

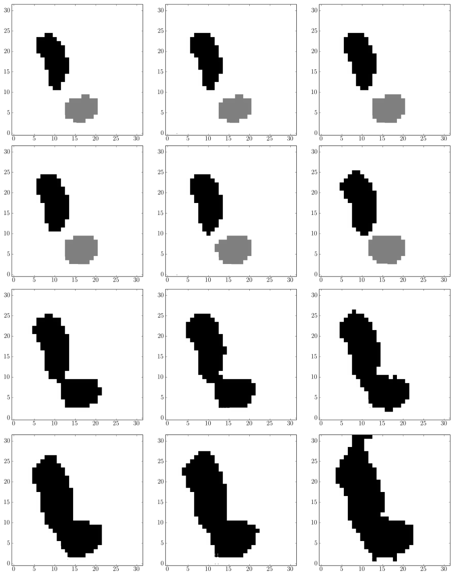


Figure 10.35: Varying the threshold level (in these images: starting at 0.1 and increasing by 0.1 counts), but leaving the structure size fixed (here 5), comparable features appear. Formulating the stopping criteria here would not be as straight forward as in the structure size case, because the threshold is a continuous and the distribution of pixel values does not lead to an a priori clear step size.

10.6.10 A Look at the Galactic Center with LIRA

Given the past tests, LIRA can be applied with some confidence to real observations. Before applying it to the relatively unknown source HESS J1837-069, the well known and well analysed region around the Galactic center was chosen as a test object for a first test on real H.E.S.S. data.

A H.E.S.S. analysis of the Galactic center region was published in Nature (Aharonian et al. 2006). The analysis in the very high energy γ-ray range clearly shows two strong point-like sources: the position of HESS J1745-290 is compatible with Sagittarius A* and Sagittarius A East. The second source G0.9+0.1 is a supernova remnant / pulsar wind nebula. In the paper it was shown that the region, after subtracting the two apparent sources, contains extended emission of γ-rays in the order of 2° in the galactic plane. The extended emission is spatially correlated with a giant molecular cloud.

The approach of subtracting the known point-like sources and the interesting results makes it a very good candidate for a first LIRA test on a real H.E.S.S. observation.

To reproduce this procedure with LIRA, two alternative ways can be followed: the positions of the two point-like sources can be taken from other multiwavelength observations,

thereby defining a physical source model. Or LIRA itself could be used in a first analysis to retrieve the most likely positions of the point sources as well as the background level. Then LIRA can be used in a second analysis to model the extended emission without the point sources, which are now included in the null-model together with the most likely background scale. This second way was chosen because it better shows LIRAs capabilities.
The H.E.S.S. observations of the galactic center region, centered on the fainter source G0.9+0.1, consist of 429 runs with a total live time of 182.885 hrs from March 2004 to September 2009. The excess map, calculated by HAP with the ring background maker and a binning of $0.02°$, can be seen in Figure 10.36. The LIRA analysis is however done with the 'on' map, the pure counts after shape cuts.

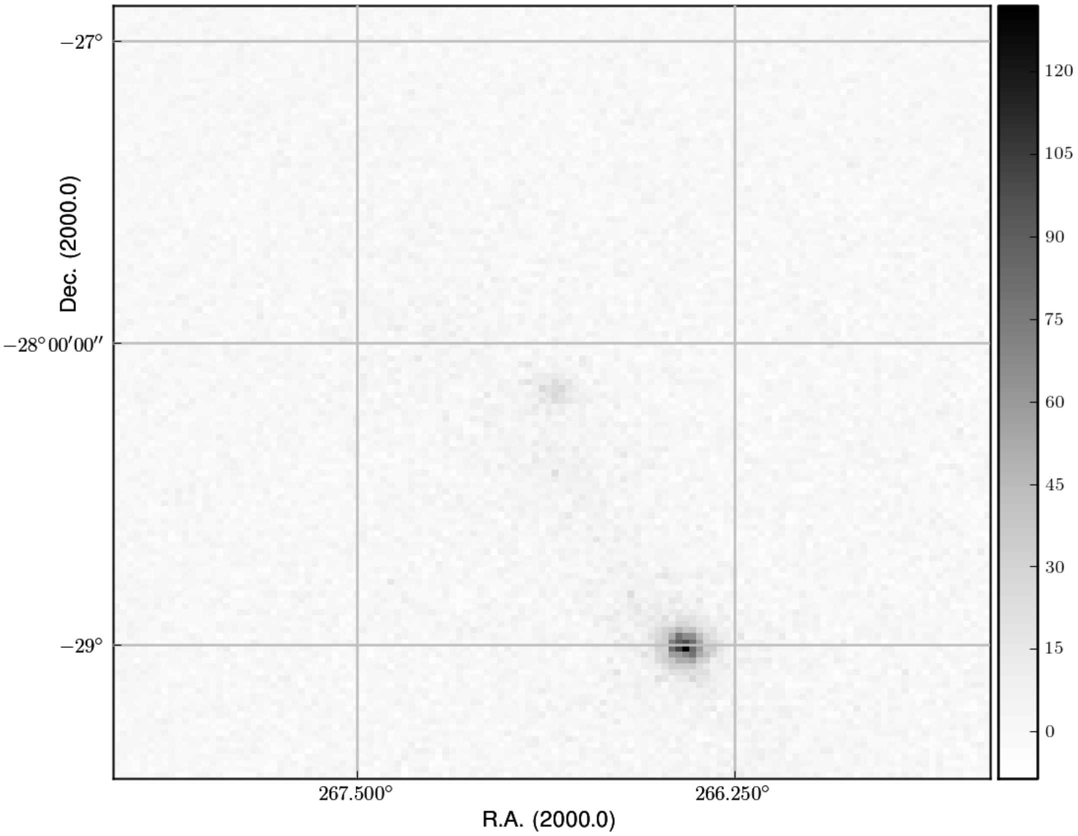


Figure 10.36: The excess map calculated by a HAP analysis of the galactic center region, centered on G0.9+0.1.

The first part of the LIRA analysis of the Galactic center region was a run with a flat background. The results shown in Figure 10.38 are gained from five independent chains with 1000 draws each. A burn in of 250 and a thinning of 15 was applied. The background, a flat image with a value of 115.80 counts/pixel[1], was fitted to be scaled by 0.103 (see Figure 10.37) which means that the reconstructed background is 11.97 counts/pixel. The multiscale component which was fitted in the same LIRA runs can be seen in Figure 10.38.

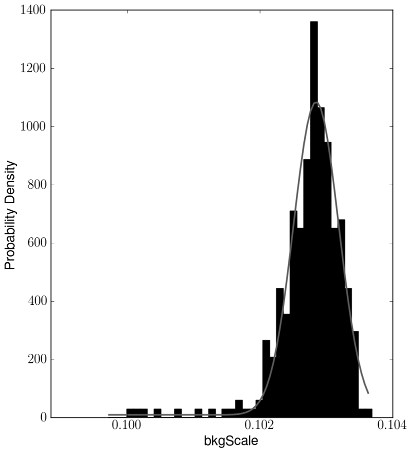


Figure 10.37: The background scale factor was fitted to 0.103. The scaled background is a flat image with 115.80 counts/pixel.

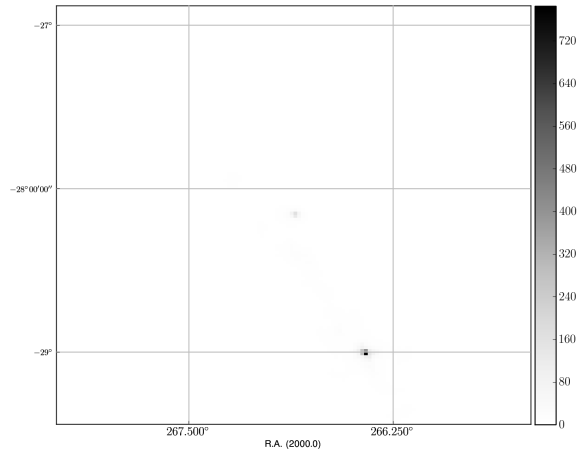


Figure 10.38: The mean image of 250 draws of the multiscale component. The draws are taken from five independent chains with 1000 samples each, a burn in of 250 and a thinning of 15. The two point sources can clearly be seen.

To create the new combined null-model, models with a point-source at the center of each hot-spot were fitted in separate LIRA runs. The scaled source models were then combined with the previously determined flat-background of 11.97 counts/pixel. This combined model was then used as the to-be-scaled null-model in another LIRA analysis. It is important to let LIRA scale the 'known' source model instead of fixing it. This can show wrong assumptions in the model. In order to find them, one has to check that the background scale factor is around 1, which it should be in the ideal case. Another way of finding wrong assumptions is to look at the resulting mean image and see if the previous empty regions are now filled with noise whereas the known sources are completely white spots. This happens when the point sources have the wrong intensity compared to the flat background. This effect can be seen in Figure 10.39. For testing purposes, a null-model was created which contained too intense point-sources, that means that the shape of the background is wrong. The LIRA algorithm could no put many events in the null-model, because this would have overemphasized the point sources, and therefore nearly all counts are placed into the extended multiresolution model.

[1] The exact value of the initial background shape is not important, however it is given here to show all input values and to be able to make sense of the scale factor, 0.103, which must be multiplied with the initial background to get the reconstructed background.

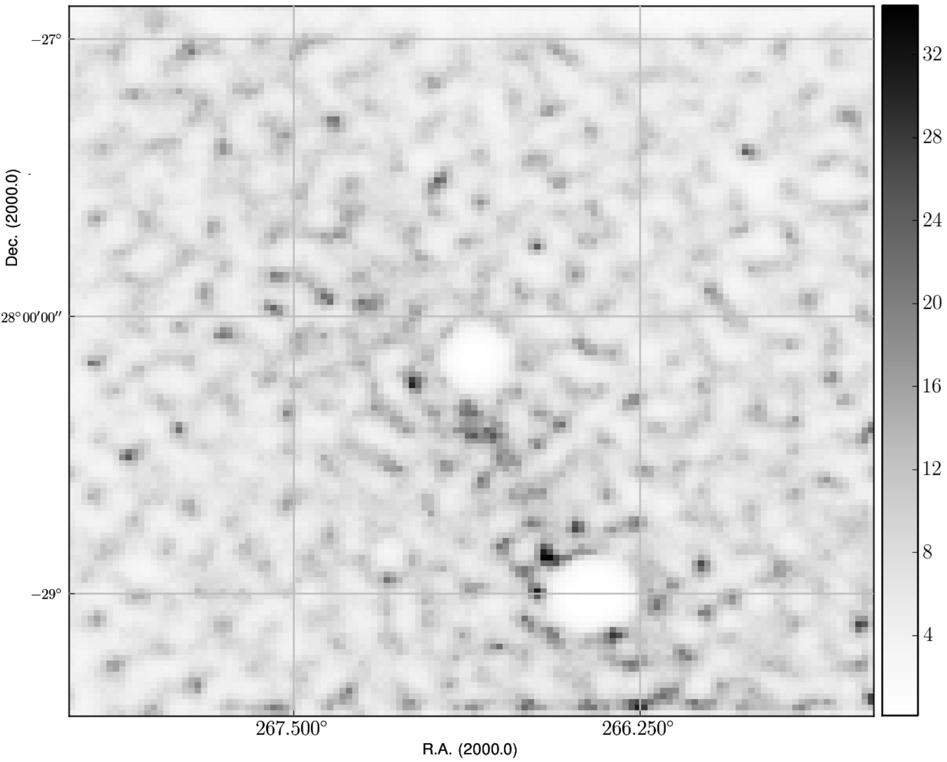


Figure 10.39: The result of a wrong null-model fit. The algorithm can not place counts in the null-model, because this would overemphasize the point sources. Therefore the flat background regions is be filled with noise.

But what happens if the point sources are underestimated compared to the background? The answer is easy for isolated point sources that only ‘sit’ on pure background: at the position of the point source will be additional, reconstructed emission. If the point source ‘sits’ on an extended emission region it is impossible to find out which of the counts belonged to the point like source and which belonged to the extended source without prior knowledge about the extended source. What is commonly done however, is to assume that the extended source will not have any extrema at the position of the point source and therefore the resulting ‘subtracted’ image should be smooth. This can be checked in the LIRA posterior mean as well.

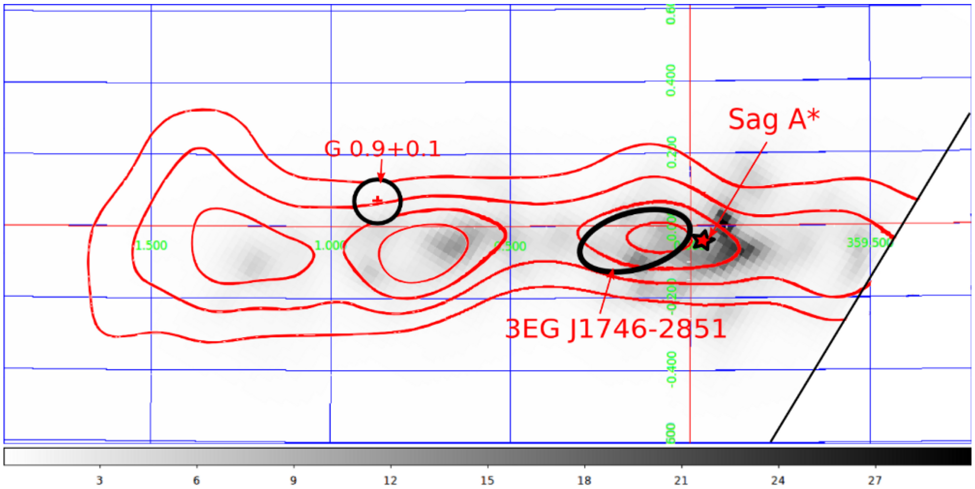


Figure 10.40: The galactic center region analysed with LIRA. The two apparent point sources have been modeled together with the background as the null-model. The residual of this null-model is shown with density contours of a giant molecular cloud as published in Aharonian et al. (2006).

The resulting image of the extended emission without the two point sources can be seen in Figure 10.40. Overlayed are the shape of the giant molecular cloud as published in Aharonian et al. (2006), the position of SNR G 0.9+0.1 and the 95% confidence region of the position of the unidentified EGRET source 3EG J1746-2851 (shown as black lines). Sagittarius A^* is marked with a black star. Like in the original paper, the visible extended emission matches the molecular cloud shape nicely. But unlike the original image, the extended emission shows smaller features which come from the deconvolution feature of LIRA.

Thus, the short look at the Galactic center showed, how known sources can be modeled with this program and how it looks like if the modelling goes wrong. This gives us some control possibilities when further extending our testing of LIRAs capabilities extending to real H.E.S.S. data of sources, which are, in contrary to the Galactic center, relatively unknown.

Chapter 11

HESS J1837-069 Analysed with LIRA

After having studied the ability of LIRA to reconstruct images of extended sources which could be expected from astronomical sources, and after showing the results of point source subtracted weak emission in the Galactic center region, the next step is to apply LIRA to HESS J1837-069. Unlike in the previous examples, this analysis will be described in more detail, especially the steps which ensure that the LIRA parameters converge and the results are unbiased samples of the posterior distributions.
The LIRA software uses a Gibbs sampler (see Section) which draws samples from a distribution. While going from one step to the next, the chain tries to optimize the distribution to better approximate the target posterior distribution. The output of a Markov chain, and in this case LIRA, are the draws of that chain. Taking all these draws for inference will lead to wrong results. At least three additional steps – burn in, thinning, convergence check – are required after LIRA runs and before the draws can be used. All these steps have already been mentioned before, because they played a role in previous chapters but they haven't been explained properly. In the following, they will be described in detail and their application within the analysis of HESS J1837-069 will be shown.

11.1 Thinning

First of all, when the Markov chain goes from one step to the next, the two steps are correlated. This means that although the draws can be considered coming from the posterior distribution, the inference from a sequence of correlated draws is less precise than from independent draws. Several solutions to this problem exist: either the simulations can be run much longer, or a chain with less correlation between the steps can be constructed (if such a chain is known), or steps in the chain can be skipped during inference (because after a number of steps, the correlation can be considered to have vanished). How many steps it takes is not known a priori, but it can be estimated with a first initial shorter test run.
This last procedure, called *thinning*, was done in all the LIRA analyses described in this thesis. For HESS J1837-069, the thinning length was determined by calculating the point where the auto-correlation $R(j) = \sum_n x_n \, \overline{x}_{n-j}$ of the various smoothing parameters x dropped to $1/e$ (see Figure 11.1). It can be seen, that a thinning of ~ 30 should be enough, that means that only every 30th draw is further processed.

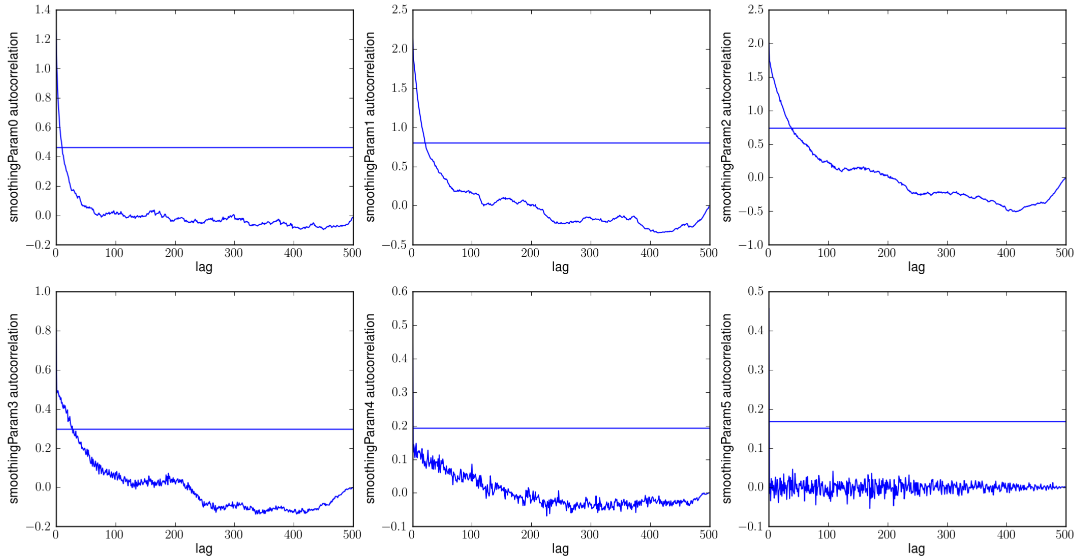


Figure 11.1: Auto-correlation of the smoothing parameters.

11.2 Burn In

The Gibbs sampler of LIRA is constructed to have the stationary distribution equal to the posterior distribution, but this does not mean, that the stationary distribution is reached right at the beginning of the chain. Checking whether and when the chain reaches its stationary state is therefore the second important step after LIRA runs. Figure 11.2 shows a scatter plot of two parameters. In the left lower corner, the chain reaches its stationary state which becomes visible as the individual steps are not recognisable anymore. Instead we see a cluster of lines. It is (in theory) possible, that a chain alternates between different stationary distributions or modes. A phenomenon like this could be easily identified in such a plot.[1]

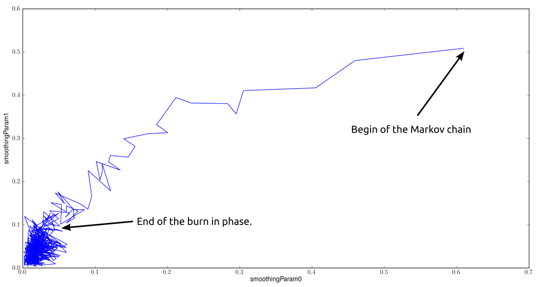


Figure 11.2: Scatter plot: Visualization of the burn in period of two parameters.

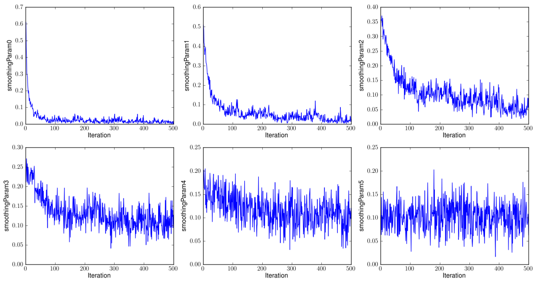


Figure 11.3: Traces of the smoothing parameters.

[1] While different modes are possible in theory, during this thesis this behaviour did not occur. Therefore no image of a real Markov chain with different modes can be shown.

The length of the burn in period, however, can not be easily accessed by looking such a scatter plot. Therefore another way of accessing the burn in length is to look at the *traces* of parameters. These traces are shown in Figure 11.3. From this figure, it can be seen, that after ~ 100 draws, the parameters are in a stable state. For the analysis of HESS J1837-069 a burn in length of 250 was chosen just to be sure that the whole burn in period is removed.

11.3 Convergence

The burn in length of the Markov chain can be considered a convergence test for a single chain, but it does not guarantee, that the chain has reached the 'global maximum likelihood'. Although there is no way to proof that the posterior distribution is reached, it can be checked, if several chains with different starting parameters converge to the same distribution. There are very sophisticated ways of assessing convergence described in Gelman et al. (2003). A very simple way is to have a look at the traces of several chains, as can be seen in Figure 11.4 for the values of the expected number of counts in the multiscale component.

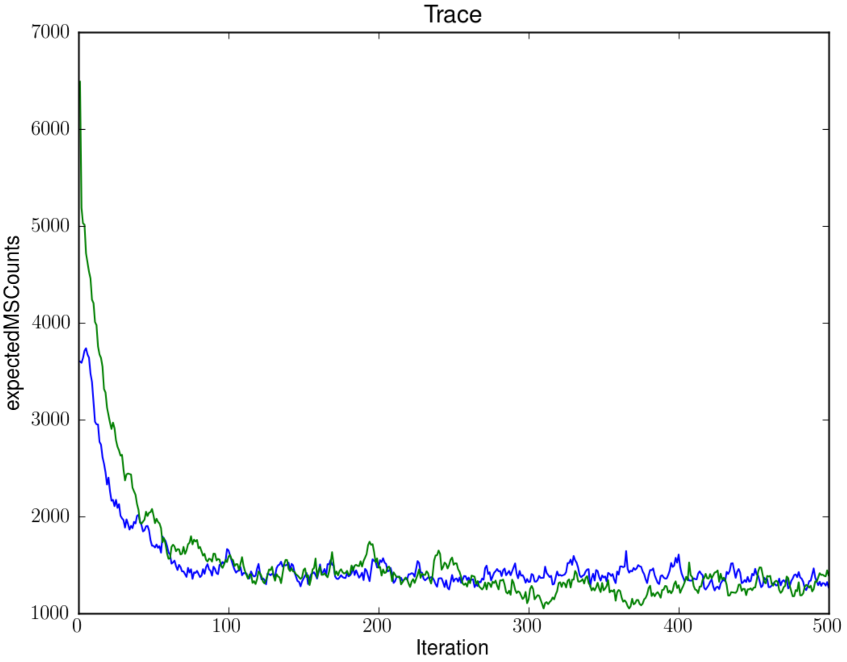


Figure 11.4: Traces of multiple chains. Convergence of both chains is reached after

The image shows the following: two different chains start at different values of the 'expectedMSCounts' parameter. One starts at 3800 counts and the other at 6500 counts. Still, they both lead to a common stationary distribution within 250 draws. Thus, a burn in length of 250 seems to be long enough.

Mean of the Multiscale Component

After removing the burn in and thinning of the draws, the mean of the reaming draws of the multiscale model can be used as a 'best fit' for the reconstructed image. This can be seen in Figure 11.5.

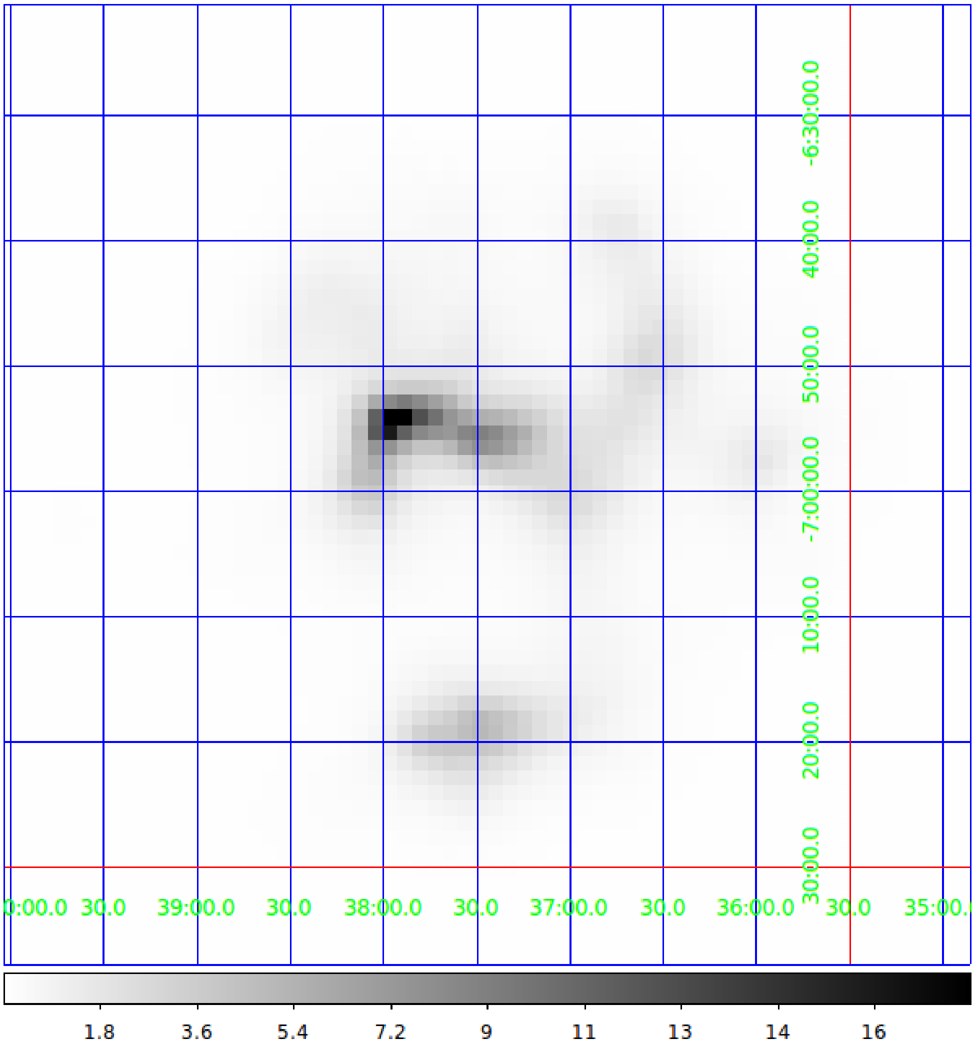


Figure 11.5: The mean of the multiscale models as a 'best fit' of the source region.

11.4 *The Vision Model applied to HESS J1837-069*

In Section 9.3 the source region of HESS J1837-069 was introduced and it was explained, that Marandon et al. (2008) claimed the existence two hot-spots in the source region. One of the questions which originally started the LIRA analysis of HESS J1837-069 within this thesis was, if the second hot-spot is significant and to what degree. Given the reconstruction technique used for this analysis, it can also be asked, where the second hot-spot is located and if it is an isolated spot or connected with other emission regions.

If the steps of the vision model from Section 10.4.1 are applied to HESS J1837-069, the result is the following Figure 11.6.

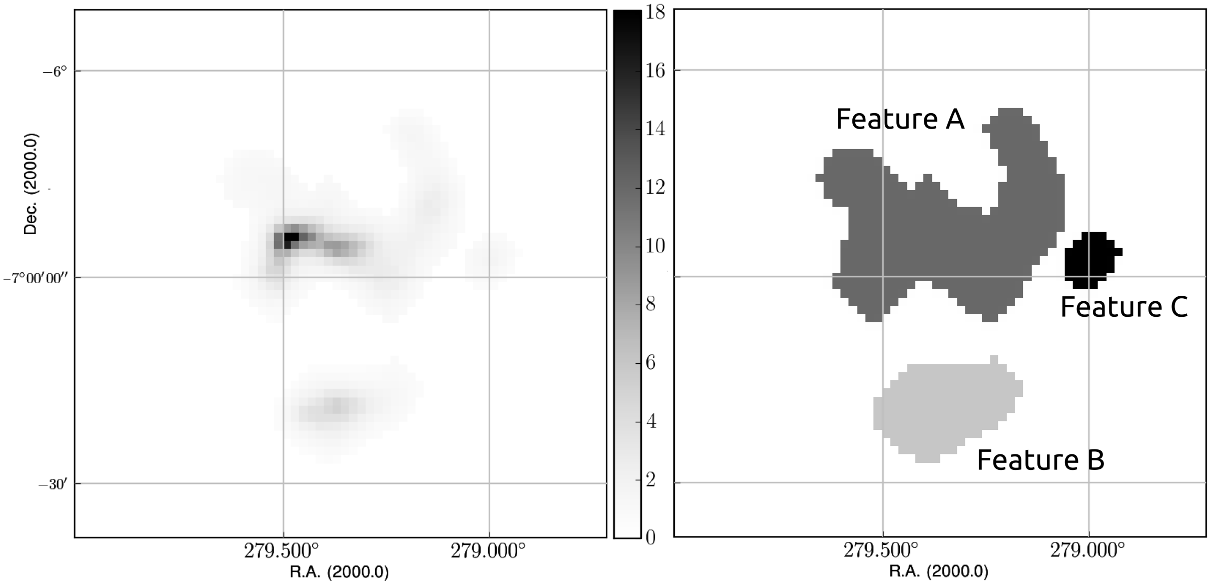


Figure 11.6: Objects in the LIRA posterior distribution as determined by the vision model described in Section 10.4.1. The number of counts is color-coded in the left image, in the right image, the colors are used to show the different features.

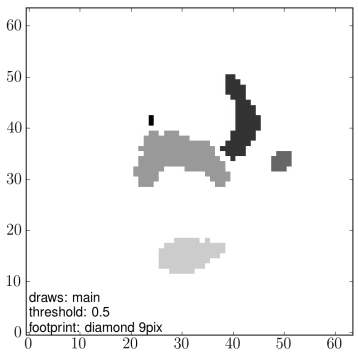


Figure 11.7: Objects in the LIRA posterior distribution as would be determined by the vision model if the features would be declared earlier (e.g. when only a 10% increase would be considered).

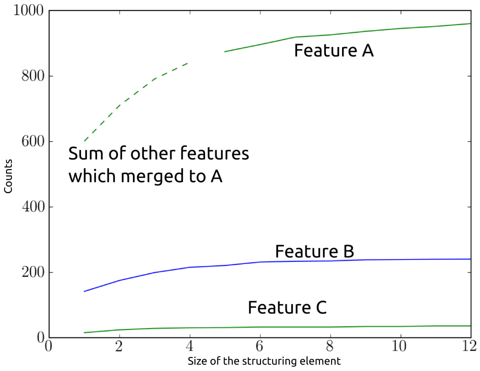


Figure 11.8: Sum of the features in the different labeled regions. Since the threshold of 1% increase is not satisfied, the main feature increases and incorporates the other smaller features.

Compared to the excess map from the HAP analysis, the numbers are smaller. The sum of all pixel values in the multi-scale model is 1551 in contrast to the sum of the pixel

Table 11.1: Main parameters of the features, based on the reconstructed image. Combined counts of the features are 1235.8. Number of counts in the complete multiscale model is 1551.1. The difference between both numbers is based on the thresholding procedures of the vision model. The significance based on Equation 17 from Li and Ma (1983) is given as a reference, the significance defined by LIRA is given in the next paragraphs.

Feature	Feature Counts	Area [pixels]	Backgr. Counts	Sign. after Li&Ma83
A	960.0	492	1151.3	16.9
B	240.0	168	393.1	7.5
C	35.8	40	93.6	2.4
Full MS model	1551.1	4096	9584.6	10.78

values in the HAP excess map, which is 2758. It must be clear however, that there are conceptual differences in both maps. The HAP excess map can not be directly interpreted as a source distribution, but is in its concept a measure of deviation from the background – positive or negative (e.g. through absorption) and therefore treats the number of counts as coming from normal distributions which can be subtracted, whereas the LIRA multi-scale model is a source model with the corresponding mathematical treatment of the poisson nature of the observation. If this is the only explanation for this difference in the number of counts can not be clarified in the time frame of this thesis. But since the simulated sources in the previous chapters showed that the number of counts in the sources and backgrounds are very well reconstructed by LIRA, this discrepancy might indeed be an artifact of the different concepts.
Main parameters of the features are summarized in Table 11.1.

The significance of the complete multi-scale model is lower than feature A, because without the boundaries defined by the vision model, the full image has to be taken as source region, which increases the number of background counts, but not the source counts.
The LIRA significance of the full multi-scale model can be calculated based on the steps described earlier. The null-model in that case is the pure background as determined in the original analysis. Unlike with the other reference analyses, the distribution of counts in the multi-scale model does not follow a normal distribution. Instead, the number of counts is dominated by values close to zero. An exponential distribution was therefore used for the fit and the OVL calculation (see Figure 11.9).

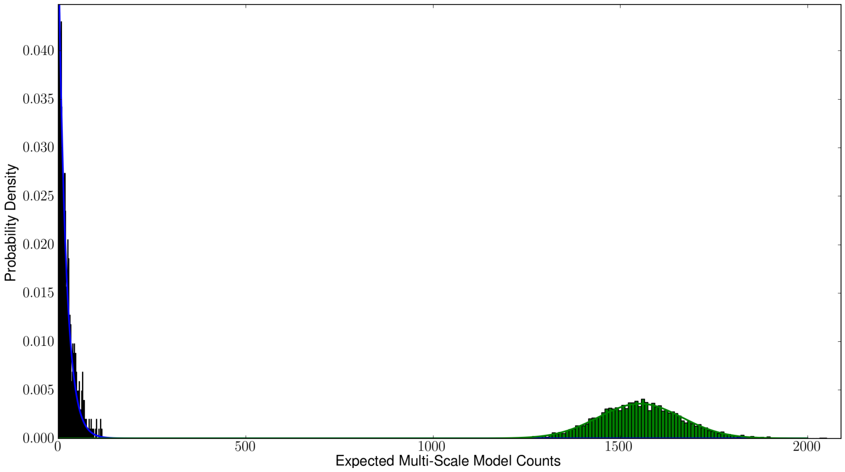


Figure 11.9: Probability distribution of the number of counts in the multiscale model for the original (green) and the reference analysis (blue).

The probability was calculated to $p = 3.266 \times 10^{-15}$ or 7.88σ. This is much lower than the 10.78σ calculated with the Li&Ma formula. However, as will be seen in the other calculations for the feature significance, this seems to be a general trend in the results based on the LIRA analysis.

11.5 The Significance of the Second Hot-Spot

As described in Section 10.4.1, the significance of a feature is determined by comparing a summary statistic of two analysis: One containing all features and one missing the feature of interest. The individual features were automatically determined by the vision model and cut from the reconstructed image of the original analysis to gain I_{cut} for the reference analysis (see Figure 11.10 for the first feature).

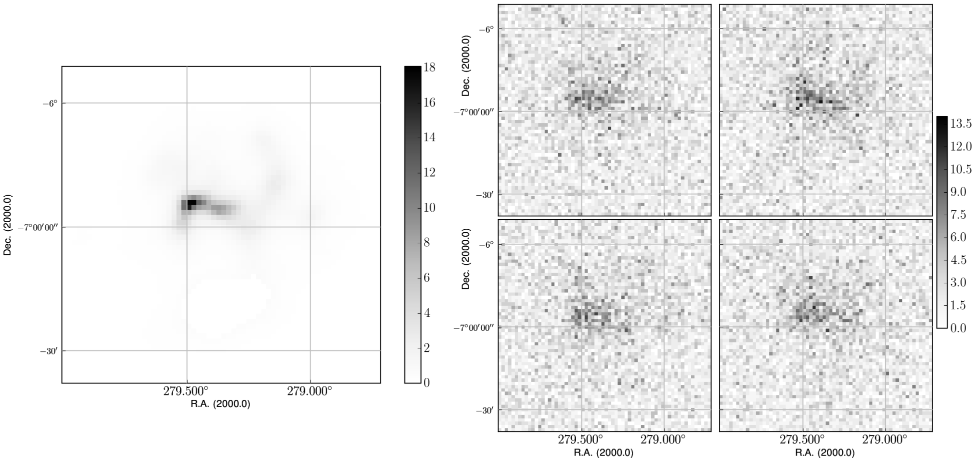


Figure 11.10: Residual image I_{cut} after cutting region B from the reconstructed image and four simulated observations.

Four observations were simulated of the new hypothetical source structure $N = B * s + I_{cut}$. These simulated observations $O_{x,y \sim} Poisson(N_{x,y})$ take the exposure and the PSF into account and are used for a reference LIRA analysis. From both LIRA outputs (the original analysis and the reference analysis) the same summary statistics were extracted and the OVL was calculated.

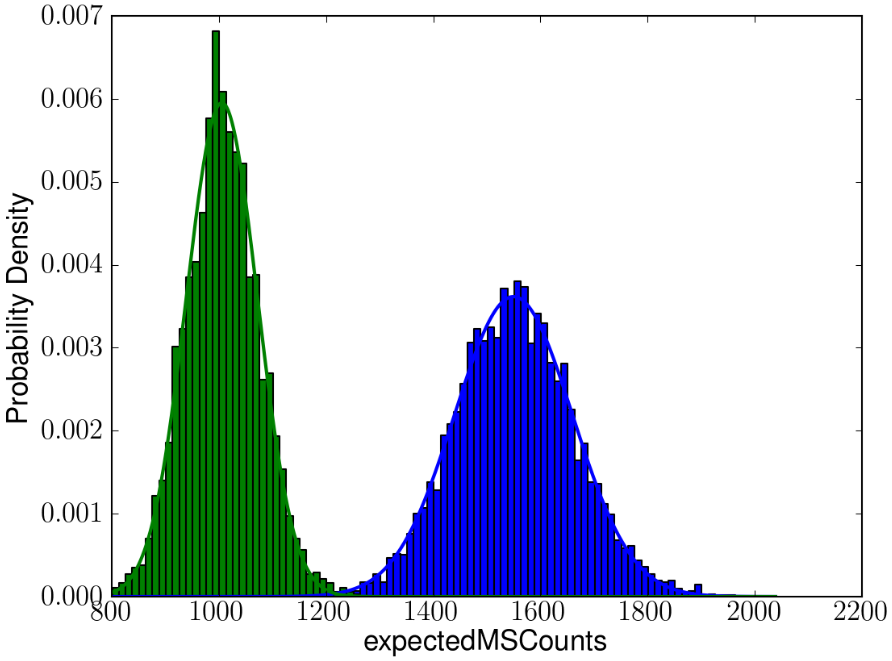


Figure 11.11: Probability distribution of the number of counts in the multiscale model for the original (blue) and the reference analysis (green). Both distributions are fit with a normal distribution.

The result of this comparison is the feature wise significance of the second hot-spot: The probability, that the second hot-spot is a fluctuation of a flat background or the surrounding source structures is $p = 0.00237$ or in other words, the hot-spot is significant at the 3.04σ level (using a two-sided Normal test). However, this second hot-spot B – if accepted as more than a fluctuation of the background – is clearly separated from the remainder of the emission region. In this regard, the claim of a second source concluded from the HAP based slice analysis described at the beginning of Chapter 9.4 could be confirmed: region B clearly is separated from the main area.

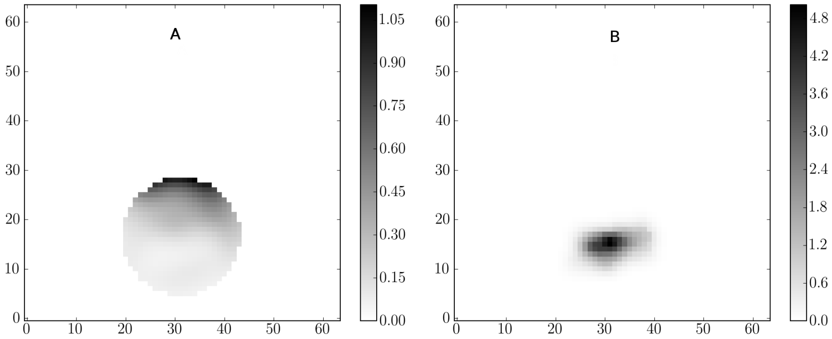


Figure 11.12: *(A)* Counts taken into account into the significance map from the surroundings of region B. The sum of counts not coming from the identified southern region B is 112.09. *(B)* The number of counts in the identified region B itself is 240.0.

Note, that the maximal **detection significance** at the position of region B as shown by the HAP significance map ($\sim 11\sigma$) is not comparable with the **feature-wise significance** because it includes not only counts from the region B itself. Notably, it also includes counts from the main region, see Figure 11.12. Therefore it highly overestimates the true significance value. Thus it is not surprising that the LIRA analysis produces lower significance numbers.

11.6 The Significance of the Main Emission Region

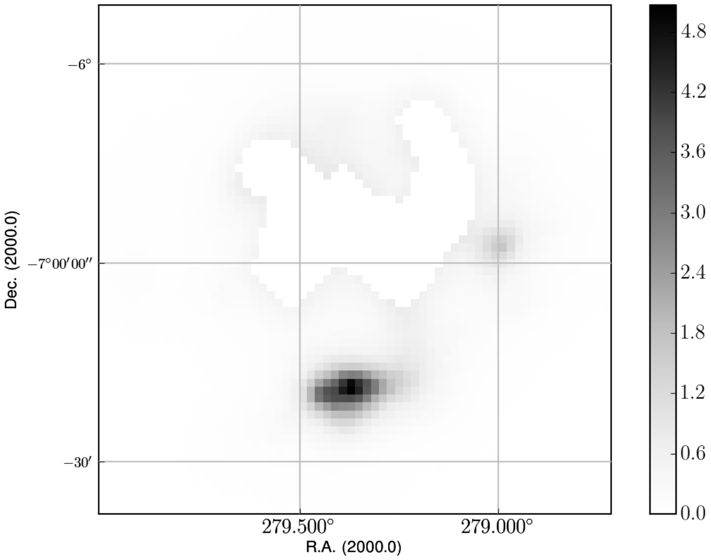


Figure 11.13: Residual image I_{cut} after cutting region A from the reconstructed image.

The significance of the region is accessed in the same way as for the second hot-spot. The distributions of the multi-scale counts for both analysis results can be seen in Figure 11.14. The main emission region was treated with the same analysis procedure as described in the last section. I_{cut} and the corresponding simulated observations can be seen in Figure 11.13.

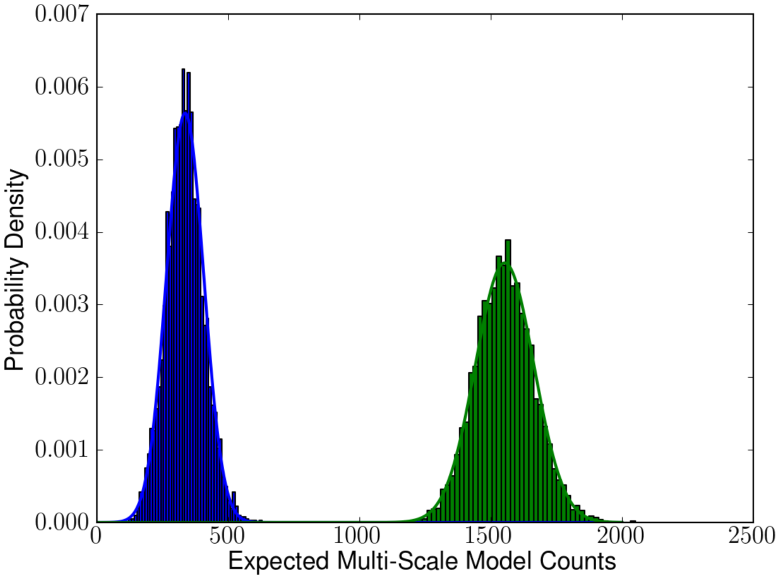


Figure 11.14: Probability distribution of the number of counts in the multiscale model for the original (green) and the reference analysis (blue). Both distributions are fit with a normal distribution.

With a probability of $p = 2.495 \times 10^{-11}$ the main region A is only a fluctuation of the background and the surrounding source regions, which corresponds to a feature significance of 6.67σ.

The third and smallest feature C is not significant ($p = 0.224$) as can be seen in Figure 11.15. An arc-like region in the main emission region can not be automatically separated into a feature. And apart from region A, the main hot-spot, no region is significant enough to definitely claim a second source.

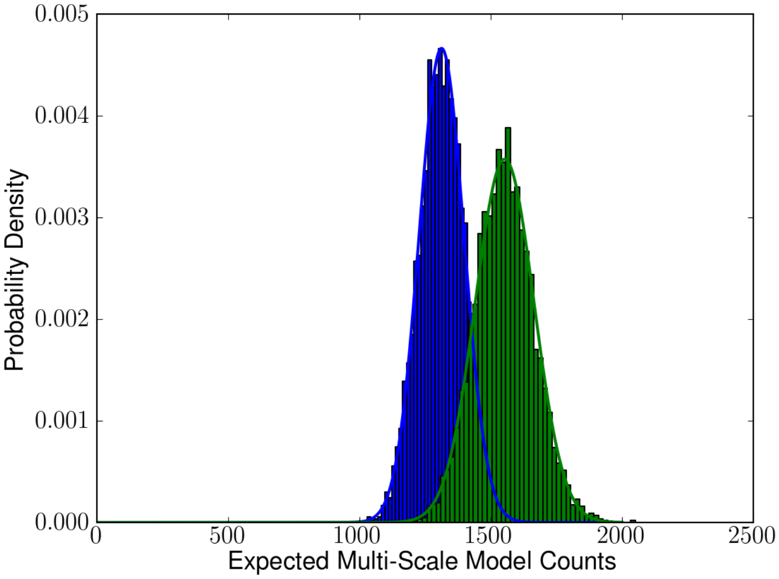


Figure 11.15: Probability distribution of the number of counts in the multiscale model for the original (green) and the reference analysis (blue). Both distributions are fit with a normal distribution.

Open questions

The reasons for this difference in the significance have not yet been identified. To find an answer to this question, a systemic study of the distribution of significances would be needed to show that the overlapping coefficient interpreted as a probability is statistically sound. Especially, it would have to be shown that for a given null-model, the reference analyses of simulations of exactly this null-model (no features removed) yield distributions of significances consistent with 0σ. And it would also be necessary to show that if features are detected from the null-model simulations, that the respective significances are small. Both simulations would be relatively easy to design, but would require a large amount of time, which can not be done for this thesis.

Another question which needs to be addressed is whether the choice of the summary statistic is good enough or if better measures could be used to increase the sensitivity of the procedure. In general, the summary statistic needs to capture the multi-parametric distance of the different posterior distributions. Since the influence of the hyperprior distribution on the number of counts in the multi-scale model seemed negligible but their influence on the smoothing parameters are not known and the number of counts is in general a good candidate for the summary statistic, no further investigation of other summary statistics was conducted for the analyses in this thesis. Once a good solution for the hyperprior problem is found, however, this question should be re-addressed in more detail. Again, this would exceed the time frame of this thesis and is therefore left for future work.

The third big open question concerns the feature separation technique. The tests on the simulated sources seem representative for the kind of source shape which can be seen in HESS J1837-069. Therefore, the first task, to create confidence in the results of the vision model applied to this specific source is fulfilled. However, the tests are not representative for a full blind feature separation, where source shapes, positions, and strengths can vary over a wide range of values. General and extensive tests of the ability to separate sources for different background levels, source distances, relative source intensities, and source shapes are very time consuming and are therefore left out for this thesis but would be a desired next step to establish this vision model as a standard analysis tool.

Another open question was already mentioned earlier: LIRA is in it's current implementation not able to include different exposure maps (where exposure is in this context meant as a combining term for the instrumental response) – one for the γ-rays and one for hadronic background – which is needed for the imaging Cherenkov technique. Changing LIRA in such a fundamental way could also not be done in addition to the conducted tests in the time of this thesis, therefore it must be left for future work.

Chapter 12

Conclusions

All the tests in the Sections 10.6.5 to 10.6.10 were run with the modified version of LIRA which sets the smoothing parameter at the pixel level to 1 (note again, that the spilt probability itself is still fit). This value is used because it marks the divide between equal splits on the one hand and splits with an emphasis on one partner on the other hand. It was shown that the resulting images are better suited for extended sources than the images created by the original LIRA software. During the thesis, I was not able to implement a better hyperprior function which would be less restrictive than setting the parameter directly. However, the significance results of the LIRA analysis should not be affected by the modified software version: The difference in the reconstruction of the prior distribution between the original and the modified version can be described as either placing counts in point sources or in smoother extended features. Both are valid prior distributions which could have lead to the same observation result. The actual number of counts which contribute to a feature should not change between the two versions. As was described earlier, the feature-wise significance is determined by a summary statistic. The obvious choice for a summary statistic is the number of counts in the extended multiscale model. And this number is identical in the original and the modified version of LIRA. So, the feature-wise significance is not affected by the modifications.

The tests with the LIRA software showed great potential for morphological analysis of low-count images. The availability of the sampled posterior distribution increases the range of accessible information and allows to define e.g. feature-wise significances. This is not possible for low count images with the standard HAP analysis. The systematic tests showed that LIRA is able to reconstruct point sources and extended sources, both in position/shape and in intensity. Backgrounds are modeled accurately even in the low-count regime. In combination with other techniques, LIRA provides the basis for removing low-count point sources and low-count extended sources from images and therefore enables the researcher with mathematically correct source ‘subtracted’ data.

But although LIRA has many advantages, questions remain in regard to the influence of the hyperprior and the best hyperprior function for H.E.S.S. sources. Due to the ill-posed nature of deconvolution in general, it is not possible to define a hyperprior function which can decide whether an observation was composed of point sources or extended emission in general.

With regard to the H.E.S.S. source HESS J1837-069 analysed in this thesis, some results can be summarized as well: Of special interest was the question, if the morphology of the source suggests a second hot-spot in the south, or if there is a better characterization for the visible phenomenon. Although it is not possible to distinguish between two sources on the one hand and two separate hot-spots of one single source on the other hand (given the time and energy integrated flux) it is still possible to claim separate emission regions. Strictly speaking, this claim is independent from a possible underlying connection between the spots. The LIRA analysis in combination with the automatic region selection shows that several features can indeed be separated, including the second hot-spot in the south. A detailed analysis, however, revealed that this hot-spot is not significant enough ($\sim 3\sigma$) to definitely claim a second source.

Taken all together, LIRA is a promising extension to HAP analyses. This thesis offered first insight into the potential as well as limitations of this analysis method. Occurring problems and questions regarding LIRA during the tests with known sources were communicated to and discussed with the LIRA developers. Thus this thesis contributed to further improving the LIRA software which will hopefully lead to a successful application within the H.E.S.S. community in the future.

Bibliography

Aharonian, F., Akhperjanian, a. G., Aye, K.-M., Bazer-Bachi, a. R., Beilicke, M., Benbow, W., et al. (2005). A new population of very high energy gamma-ray sources in the Milky Way. *Science (New York, N.Y.)*, 307(5717):1938–42.

Aharonian, F., Akhperjanian, A. G., Aye, K.-M., Bazer-Bachi, A. R., Beilicke, M., Benbow, W., et al. (2004). Very high energy gamma rays from the direction of Sagittarius A*. A&A, 425:L13–L17.

Aharonian, F., Akhperjanian, A. G., Bazer-Bachi, A. R., Beilicke, M., Benbow, W., Berge, D., et al. (2006a). Observations of the Crab nebula with HESS. A&A, 457:899–915.

Aharonian, F., Akhperjanian, A. G., Bazer-Bachi, A. R., Beilicke, M., Benbow, W., Berge, D., et al. (2006b). First detection of a VHE gamma-ray spectral maximum from a cosmic source: HESS discovery of the Vela X nebula. A&A, 448:L43–L47.

Aharonian, F., Akhperjanian, A. G., Bazer-Bachi, A. R., Beilicke, M., Benbow, W., Berge, D., et al. (2006). Discovery of very-high-energy gamma-rays from the galactic centre ridge. *Nature*, 439(7077):695–8.

Aharonian, F., Akhperjanian, A. G., Bazer-Bachi, A. R., Beilicke, M., Benbow, W., Berge, D., et al. (2006). Energy dependent γ-ray morphology in the pulsar wind nebula HESS J1825-137. A&A, 460:365–374.

Aharonian, F. A., Atoyan, A. M., and Kifune, T. (1997). Inverse Compton gamma radiation of faint synchrotron X-ray nebulae around pulsars. MNRAS, 291:162–176.

Aharonian, F. A. and Bogovalov, S. V. (2003). Exploring physics of rotation powered pulsars with sub-10 GeV imaging atmospheric Cherenkov telescopes. New A, 8:85–103.

Aliu, E., Anderhub, H., Antonelli, L. A., Antoranz, P., Backes, M., Baixeras, C., et al. (2008). Observation of Pulsed γ-Rays Above 25 GeV from the Crab Pulsar with MAGIC. *Science*, 322:1221–.

Anada, T., Ebisawa, K., Dotani, T., and Bamba, A. (2009). X-Ray Studies of HESS J1837-069 with Suzaku and ASCA: a VHE Gamma-Ray Source Originated from the Pulsar Wind Nebula. *Publications of the Astronomical Society of Japan*.

Baldini, L. and the Fermi LAT Collaboration (2011). Fundamental physics in space with the Fermi Gamma-ray Space Telescope. *Journal of Physics Conference Series*, 306(1):012014–+.

Baring, M. G. (2011). High-Energy Emission from Pulsars and their Systems. *Astrophysics and Space Science*, pages 453–472.

Berge, D., Funk, S., and Hinton, J. (2007). Background modelling in very-high-energy γ-ray astronomy. A&A, 466:1219–1229.

Bernlohr, K. (2000). Impact of atmospheric parameters on the atmospheric Cherenkov technique*. *Astroparticle Physics*, 12:255–268.

Bernlöhr, K., Carrol, O., Cornils, R., Elfahem, S., Espigat, P., Gillessen, S., et al. (2003). The optical system of the H.E.S.S. imaging atmospheric Cherenkov telescopes. Part I: layout and components of the system. *Astroparticle Physics*, 20:111–128.

Bernlöhr, K. (2000). Monte-Carlo images of air showers.

Bertin, E. and Arnouts, S. (1996). SExtractor: Software for source extraction. *Astronomy and Astrophysics Supplement*.

Bijaoui, A., Slezak, E., and Vandame, B. (1998). A Multiscale Vision Model and Applications to Astronomical Image and Data Analyses. *"Astrophysics and Algorithms: a DIMACS Workshop on Massive Astronomical Data Sets"*.

Bolz, O. (2004). *Absolute Energiekalibration der abbildenden Cherenkov-Teleskope des H.E.S.S. Experiments und Ergebnisse erster Beobachtungen des Supernova-Uberrests RX J1713.7-3946*. PhD thesis, Ruprecht – Karls – Universität Heidelberg.

Borla Tridon, D., Goebel, F., Fink, D., Haberer, W., Hose, J., Hsu, C. C., et al. (2009). Performance of the Camera of the MAGIC II Telescope. *ArXiv e-prints*.

Bradt, H. V., Rothschild, R. E., and Swank, J. H. (1993). X-ray timing explorer mission. A&AS, 97:355–360.

Brinker, T., Degenhardt, H., and Kupris, G. (2007). *Embedded Linux*. Vde Verlag GmbH.

Calabretta, M. R. and Greisen, E. W. (2002). Representations of celestial coordinates in FITS. A&A, 395:1077–1122.

Chaves, R. C. G., de Oña Wilhemi, E., and Hoppe, S. (2008). The H.E.S.S. Galactic Plane Survey. In F. A. Aharonian, W. Hofmann, & F. Rieger, editor, *American Institute of Physics Conference Series*, volume 1085 of *American Institute of Physics Conference Series*, pages 219–222.

Coifman, R. R. and Donoho, D. L. (1995). Translation-Invariant De-Noising. pages 125—-150. Springer-Verlag.

Connors, A., Stein, N. M., van Dyk, D., Kashyap, V., and Siemiginowska, A. (2011). LIRA – The Low-Counts Image Restoration and Analysis Package: A Teaching Version via R. In Evans, I. N., Accomazzi, A., Mink, D. J., and Rots, A. H., editors, *Astronomical Data Analysis Software and Systems XX*, volume 442 of *Astronomical Society of the Pacific Conference Series*, page 463.

Connors, A. and van Dyk, D. A. (2007). How To Win With Non-Gaussian Data: Poisson Goodness-of-Fit. *Statistical challenges in astronomy Third Statistical Challenges in Modern Astronomy*, 371:101–117.

Cornils, R. (2006). *Justierung und Abbildungsfunktion der H.E.S.S.-Reflektoren sowie Untersuchung der ultraleuchtkräftigen Infrarot-Galaxie Arp 220 mit dem H.E.S.S.-Teleskopsystem.* PhD thesis, Universität Hamburg.

Cornils, R., Bernlöhr, K., Heinzelmann, G., Hofmann, W., and Panter, M. (2005). The optical system of the H.E.S.S. II telescope. In *International Cosmic Ray Conference*, volume 5 of *International Cosmic Ray Conference*, pages 171–+.

CTA Consortium, T. (2010). Design Concepts for the Cherenkov Telescope Array. *ArXiv e-prints.*

Daubechies, I. (1992). *Ten lectures on wavelets.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Daum, A., Hermann, G., Hess, M., Hofmann, W., Lampeitl, H., Pühlhofer, G., et al. (1997). First results on the performance of the HEGRA IACT array. *Astroparticle Physics*, 8:1–11.

de Jager, O. C. and Djannati-Ataï, A. (2008). Implications of H.E.S.S. observations of pulsar wind nebulae. *ArXiv e-prints.*

de Naurois, M. (2006). Analysis methods for Atmospheric Cerenkov Telescopes. *ArXiv Astrophysics e-prints.*

de Naurois, M. and the H. E. S. S. Collaboration (2003). Application of an Analysis Method Based on a Semi-Analytical Shower Model to the First H.E.S.S. Telescope. In *International Cosmic Ray Conference*, volume 5 of *International Cosmic Ray Conference*, pages 2907–+.

Deil, C., van Eldik, C., Förster, A., Hermann, G., Hofmann, W., and Panter, M. (2008). H.E.S.S. II-Telescope Structure, Reflector and Drive System. In F. A. Aharonian, W. Hofmann, & F. Rieger, editor, *American Institute of Physics Conference Series*, volume 1085 of *American Institute of Physics Conference Series*, pages 693–695.

Esch, D. N., Connors, A., Karovska, M., and van Dyk, D. A. (2004). An Image Restoration Technique with Error Estimates. *The Astrophysical Journal*, 610(2):1213–1227.

Figer, D. F., MacKenty, J. W., Robberto, M., Smith, K., Najarro, F., Kudritzki, R. P., et al. (2006). Discovery of an Extraordinarily Massive Cluster of Red Supergiants. *The Astrophysical Journal*, 643(2):1166–1179.

Funk, S. (2005). *A new population of very high-energy γ-ray sources detected with H.E.S.S. in the inner part of the Milky Way.* PhD thesis.

Gaensler, B. M. and Slane, P. O. (2006). The Evolution and Structure of Pulsar Wind Nebulae. *Annual Review of Astronomy and Astrophysics*, 44(1):17–47.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2003). *Bayesian Data Analysis.* Chapman and Hall/CRC.

Gotthelf, E. V., Halpern, J. P., Camilo, F., Markwardt, C., and Swank, J. (2008). Discovery of a 70.5 ms Pulsar in AX J1838.0-0655 Associated with HESS J1837-069. *The Astronomer's Telegram.*

Gotthelf, E. V. and Halpern, J. P. (2008). Pulsar Wind Nebula Identification(s) For The TeV Gamma-ray Source HESS J1837-069. *American Astronomical Society*.

Haar, A. (1910). Zur Theorie der orthogonalen Funktionensysteme. *Mathematische Annalen*, 69(3):331–371.

Harding, A. K., Muslimov, A. G., and Zhang, B. (2002). Regimes of Pulsar Pair Formation and Particle Energetics. ApJ, 576:366–375.

Heck, D., Knapp, J., Capdevielle, J. N., Schatz, G., and Thouw, T. (1998). *CORSIKA: a Monte Carlo code to simulate extensive air showers*.

HEGRA Collaboration, Heß, M., Bernlöhr, K., Daum, A., Hemberger, M., Hermann, G., et al. (1999). The time structure of Cherenkov images generated by TeV gamma-rays and by cosmic rays. *Astroparticle Physics*, 11:363–377.

Heijmans, H. and Ronse, C. (1990). The algebraic basis of mathematical morphology I. Dilations and erosions. *Computer Vision, Graphics, and Image Processing*, 50(3):245–295.

Heitler, W. (1954). *Quantum theory of radiation*.

Hertz, P. and Grindlay, J. E. (1988). The Einstein Galactic plane survey. II - Optical observations of Northern Hemisphere X-ray sources. *The Astronomical Journal*, 96:233.

HESS Collaboration, Abramowski, A., Acero, F., Aharonian, F., Akhperjanian, A. G., Anton, G., et al. (2011). Revisiting the Westerlund 2 field with the HESS telescope array. A&A, 525:A46+.

Hewish, A., Bell, S. J., Pilkington, J. D. H., Scott, P. F., and Collins, R. A. (1968). Observation of a Rapidly Pulsating Radio Source. Nature, 217:709–713.

Hillas, A. M. (1985). Cerenkov light images of EAS produced by primary gamma. In F. C. Jones, editor, *International Cosmic Ray Conference*, volume 3 of *International Cosmic Ray Conference*, pages 445–448.

Hinton, J. and Hofmann, W. (2009). Teraelectronvolt Astronomy. *Annual Review of Astronomy and Astrophysics*, 47(1):523–565.

Hirotani, K., Harding, A. K., and Shibata, S. (2003). Electrodynamics of an Outer Gap Accelerator: Formation of a Soft Power-Law Spectrum between 100 MeV and 3 GeV. ApJ, 591:334–353.

Hofmann, W. (1998). Alignment of the hess mirrors using images of stars; (internal note).

Horns, D. and the HESS Collaboration (2007). H.E.S.S.: Status and future plan. *Journal of Physics Conference Series*, 60:119–122.

IBM Corporation (2001). On-Chip Peripheral Bus; Architecture Specifications; Version 2.1.

Inman, H. F. and Bradley, E. L. (1989). The overlapping coefficient as a measure of agreement between probability distributions and point estimation of the overlap of two normal densities. *Communications in Statistics - Theory and Methods*, 18(10):3851–3874.

Jones, E., Oliphant, T., Peterson, P., et al. (2001–2011). SciPy: Open source scientific tools for Python.

Kelner, S. R., Aharonian, F. A., and Bugayov, V. V. (2006). Energy spectra of gamma rays, electrons, and neutrinos produced at proton-proton interactions in the very high energy regime. Phys. Rev. D, 74(3):034018–+.

Knuth, D. E. (1976). Big Omicron and big Omega and big Theta. *ACM SIGACT News*, 8(2):18–24.

Kohnle, A., Mattes, J., Hermann, G., Hofmann, W., and Panter, M. (2000). Photodetectors for hess. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 442(1–3):322 – 326.

Kolaczyk, E. D. and Nowak, R. D. (2000). A Multiresolution Analysis for Likelihoods: Theory and Methods.

Kolaczyk, E. D. and Nowak, R. D. (2004). Multiscale likelihood analysis and complexity penalized estimation. *The Annals of Statistics*, 32(2):500–527.

Li, T.-P. and Ma, Y.-Q. (1983). Analysis methods for results in gamma-ray astronomy. *The Astrophysical Journal*, 272:317.

Loredo, T. J. (1992). Promise of Bayesian inference for astrophysics. In E. D. Feigelson & G. J. Babu, editor, *Statistical Challenges in Modern Astronomy*, pages 275–306.

Mallat, S. (2008). *A wavelet tour of signal processing ; The Sparse Way.* Academic Press, 3rd edition.

Manchester, R. N., Hobbs, G. B., Teoh, A., and Hobbs, M. (2005). The Australia Telescope National Facility Pulsar Catalogue. AJ, 129:1993–2006.

Marandon, V., Djannati-Atai, A., Terrier, R., Puehlhofer, G., Hauser, D., Schwarzburg, S., et al. (2008). A closer look at HESS J1837-069 following the pulsar discovery. In Aharonian, F. A., Hofmann, W., and Rieger, F., editors, *American Institute of Physics Conference Series*, volume 1085 of *American Institute of Physics Conference Series*, pages 320–323.

Marshall, J. (2007). H.E.S.S. internal notes: Image processing techniques applied to H.E.S.S. skymaps.

Migliazzo, J. M., Gaensler, B. M., Backer, D. C., Stappers, B. W., van der Swaluw, E., and Strom, R. G. (2002). Proper-Motion Measurements of Pulsar B1951+32 in the Supernova Remnant CTB 80. ApJ, 567:L141–L144.

Moiseev, A. A. (2008). Gamma-ray Large Area Space Telescope: Mission overview. *Nuclear Instruments and Methods in Physics Research A*, 588:41–47.

Montanus, J. M. C. (2012). Intermediate models for longitudinal profiles of cosmic showers. *Astroparticle Physics*, 35:651–659.

Opdenacker, M. (2007). Introduction to uClinux.

Paulin, H. (2009). Invent.org.

Protassov, R., van Dyk, D. A., Connors, A., Kashyap, V. L., and Siemiginowska, A. (2002). Statistics, handle with care: Detecting multiple model components with the likelihood ratio test. *The Astrophysical Journal*, 571:545–559.

Reiser, B. and Faraggi, D. (1999). Confidence intervals for the overlapping coefficient: The normal equal variance case. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 48(3):pp. 413–418.

Reynolds, S. P. (2008). Supernova Remnants at High Energy. *Annual Review of Astronomy and Astrophysics*, 46(1):89–126.

Ritz, S. (2007). Overview of the GLAST Mission and Opportunities. In S. Ritz, P. Michelson, & C. A. Meegan, editor, *The First GLAST Symposium*, volume 921 of *American Institute of Physics Conference Series*, pages 3–7.

Rossi, B. and Greisen, K. (1941). Cosmic-Ray Theory. *Reviews of Modern Physics*, 13:240–309.

Stamatescu, V., Rowell, G., Thornton, G., Clay, R., Dawson, B., Denman, J., et al. (2008). Analysis Techniques at large core distances for multi-TeV Gamma Ray Astronomy. In F. A. Aharonian, W. Hofmann, & F. Rieger, editor, *American Institute of Physics Conference Series*, volume 1085 of *American Institute of Physics Conference Series*, pages 846–849.

Starck, J. and Murtagh, F. (2006). *Astronomical image and data analysis*. Springer.

Starck, J.-l., Murtagh, F., and Fadili, J. M. (2010). *Sparse Image and Signal Processing*. Cambridge University Press, Cambridge.

Terrier, R., Djannati-Atai, A., Hoppe, S., Marandon, V., Renaud, M., and de Jager H.E.S.S. Collaboration, O. (2008). H.e.s.s. observations of the young composite snr kes 75. *AIP Conference Proceedings*, 1085(1):316–319.

Teshima, M. (2008). MAGIC discovers pulsed VHE gamma ray emission above 25 GeV from the Crab pulsar. *The Astronomer's Telegram*, 1491:1–+.

The R Development Core Team (2011). R Manual.

Tibolla, O., Chaves, R. C. G., Domainko, W., de Jager, O., Kaufmann, S., Wagner, S., et al. (2009). New unidentified Galactic H.E.S.S. sources. *ArXiv e-prints*.

van Rossum, G. et al. (1991–2011). The Python programming language.

van Rossum, G. and Warsaw, B. (2001). Style guide for python code.

VERITAS Collaboration, Aliu, E., Arlen, T., Aune, T., Beilicke, M., Benbow, W., et al. (2011). Detection of Pulsed Gamma Rays Above 100 GeV from the Crab Pulsar. *Science*, 334:69–.

Vincent, P., Bazer-Bachi, A. R., Chounet, L.-M., Corona, P., Degerly, Y., Degrange, B., et al. (2005). A new camera for the HESS phase II experiment. In F. A. Aharonian, H. J. Völk, & D. Horns, editor, *High Energy Gamma-Ray Astronomy*, volume 745 of *American Institute of Physics Conference Series*, pages 791–796.

Weekes, T. C., Badran, H., Biller, S. D., Bond, I., Bradbury, S., Buckley, J., et al. (2002). VERITAS: the Very Energetic Radiation Imaging Telescope Array System. *Astroparticle Physics*, 17:221–243.

Weekes, T. C., Cawley, M. F., Fegan, D. J., Gibbs, K. G., Hillas, A. M., Kowk, P. W., et al. (1989). Observation of TeV gamma rays from the Crab nebula using the atmospheric Cerenkov imaging technique. ApJ, 342:379–395.

Wiedner, C. A. (1998). Site Aspects of the HESS Project Astronomical and Visibility Conditions.

Wilhelmi, E. D. O. (2011). Cherenkov Telescopes Results on Pulsar Wind Nebulae and Pulsars. In D. F. Torres & N. Rea, editor, *High-Energy Emission from Pulsars and their Systems*, pages 435–+.

Xilinx (2009). Xilinx DS099 Spartan-3 FPGA Family data sheet.

Acknowledgements

I would like to thank several people for their support:

- Prof. Dr. Andrea Santangelo
  for the opportunity to write this thesis and the support and advice given during my time at the IAAT
- Dr. Eckhard Kendziorra
  for all the help with the alignment hardware, but especially for the fantastic mentoring through all my years at the IAAT
- Dr. Gerd Pühlhofer
  for all the helpful questions and suggestions especially regarding the data analysis of 1837
- Alanna Connors
  for letting me use LIRA and supporting me whenever questions arose. For even typing answers with an insured hand!
- Thomas Schanz
  for his help with the FPGA development and computer setups
- Prof. Dr. Dieter Horns
  for introducing me to the HESS experiment
- Dr. Christoph Tenzer
  for a lot! I think we had a great time at the University together. And we should really finish that Geocaching App, it would make us rich!
- Dr. Michael Martin
  for the good times in our office, the games, the music…
- My parents
  who always supported me in every possible way
- My wife Judith
  for everything! Including the proofreading of the thesis. And watching Doctor Who with me :-)