



UNIVERSITÀ DEGLI STUDI DI TRIESTE
UNIVERSITÉ GRENOBLE ALPES

XXXII CICLO DEL DOTTORATO DI RICERCA IN FISICA

PO FRIULI VENEZIA GIULIA - FONDO SOCIALE EUROPEO 2014/2020

DEVELOPMENT OF A COMPUTING FARM FOR
CLOUD COMPUTING ON GPU

DEVELOPMENT AND OPTIMISATION OF DATA-ANALYSIS
METHODOLOGIES FOR THE CHERENKOV TELESCOPE
ARRAY

Settore scientifico disciplinare: FIS/01

Ph.D. student
Thomas Gasparetto

Ph.D. program Coordinator
Prof. Francesco Longo

Thesis Supervisors
Prof. Francesco Longo
Prof. Giovanni Lamanna

Co-supervisor
Dr. Thomas Vuillaume

Academic Year 2018/2019

Contents

Riassunto	v
Abstract	vii
Résumé	ix
1 The Cherenkov Telescope Array	1
1.1 Imaging Cherenkov telescopes	1
1.1.1 IACTs	6
1.2 CTA sites and telescopes	7
1.2.1 Large Size Telescope	9
1.2.2 Medium Size Telescope	10
1.2.3 Small Size Telescope	11
1.3 CTA Performances	13
1.3.1 Results from prod3b	14
1.3.2 CORSIKA	16
1.3.3 sim_telarray	17
1.3.4 Analyses for IRF production	18
1.3.5 IRF usage	25
1.4 CTA Computing	26
2 Science with CTA	27
2.1 Science themes	27
2.1.1 Key Science Projects	27
2.2 Gamma-Ray Bursts	33
2.2.1 Introduction	33
2.2.2 Phases of a GRB	36
2.2.3 GRB modelling: the fireball model	38
2.2.4 HE emission from GRBs	40
2.2.5 Progenitors	43
2.2.6 Prospects for CTA	44
2.2.7 Summary	47
2.3 Gravitational Waves counterparts	47

2.3.1	Introduction	47
2.3.2	Physics of GWs	49
2.3.3	Prospects for CTA	51
3	The First Data Challenge	55
3.1	Goals	55
3.2	Technical inputs	56
3.2.1	Science tools	56
3.2.2	IRFs	56
3.2.3	Physical models	57
3.2.4	The extragalactic sky	58
3.3	The dataset	63
3.4	The close-out document	64
3.5	Conclusion	65
4	Prospects for GRB and GW follow-up with CTA	67
4.1	GRB with CTA	67
4.1.1	Fermi extrapolation	67
4.1.2	GRB Consortium paper	70
4.2	ctools_pipe	75
4.2.1	Jobs submission	75
4.2.2	Folder structure	76
4.2.3	Tasks	76
4.3	Prospects for GW follow-up with CTA	80
4.3.1	Pipeline for GW follow-up	81
4.4	Conclusions	82
5	CTA reconstruction pipeline	85
5.1	ctapipe	85
5.1.1	Framework implementation	86
5.1.2	Analysis steps in ctapipe	88
5.1.3	Contribution to ctapipe	92
5.2	Reconstruction algorithms	94
5.2.1	Hillas parametrization	94
5.2.2	Model-based analysis	98
5.2.3	Template-based analysis: ImPACT	101
5.2.4	Machine Learning	109
5.3	Visualization	111
5.3.1	OpenSCAD displayer	111
5.3.2	CREED_VTK	112

6	Optimizing for computing	117
6.1	Profiling and optimizations	117
6.2	ImPACT on GPU	122
6.2.1	PyTorch implementation	123
7	Optimizing for Physics	129
7.1	Pointing modes and divergent	129
7.1.1	Goals	131
7.1.2	Previous works	134
7.1.3	Simulations and analyses	135
7.1.4	Divergent pointing in ctapipe	139
7.1.5	Results	142
7.1.6	Conclusions	145
8	Conclusions	147
A	The Extragalactic Background Light	149
B	CTAO hierarchical data model	153
	Glossary	165

Riassunto

Il lavoro di questa tesi è stato sviluppato nel contesto del Cherenkov Telescope Array (CTA), il principale osservatorio di prossima generazione per l'astronomia a raggi gamma ad altissima energia a terra. L'obiettivo di questo lavoro è l'uso di GPU e Cloud Computing per accelerare le attività di calcolo più impegnative, sviluppando e ottimizzando le pipeline di analisi dati.

La tesi si articola in due parti principali: la prima è dedicata alla stima delle future performance di CTA riguardo l'osservazione di fenomeni violenti come quelli che generano i Gamma Ray Bursts (GRB) e le onde gravitazionali, con un lavoro iniziale sulla creazione dei modelli per il Primo CTA Data Challenge. La seconda parte della tesi è legata allo sviluppo delle pipeline per la ricostruzione dei dati di basso livello provenienti dalle simulazioni di Montecarlo utilizzando la libreria software denominata `ctapipe`. Il capitolo 1 tratta nel dettaglio il progetto CTA, i telescopi e le prestazioni dell'array, insieme ai metodi utilizzati per derivarle dalle simulazioni Monte Carlo. Gli obiettivi scientifici di CTA e dei Key Science Projects (KSP) saranno trattati nel capitolo 2, con particolare attenzione ai Gamma Ray Bursts e al follow-up degli eventi di onde gravitazionali.

Il lavoro svolto per il First CTA Data Challenge (DC-1) è presentato nel capitolo 3. Più di 500 sorgenti extragalattiche sono state modellate combinando informazioni contenute in diversi cataloghi per creare una popolazione di Nuclei Galattici Attivi (AGN). Questo Challenge è stato importante sia per coinvolgere più persone nell'analisi dei dati di CTA che per calcolare il tempo di osservazione necessario ai diversi KSP. Le simulazioni per gli articoli di Consorzio relativi ai GRB e alle onde gravitazionali, sono stati creati con la pipeline `ctools_pipe` (presentata nel capitolo 4), implementata usando le librerie `ctools` e `gammalib`. La pipeline è composta da due parti principali: il compito da eseguire (simulazione in background, creazione di modelli e detection della sorgente) e centro di calcolo in cui eseguirlo.

La seconda parte della tesi è incentrata sullo sviluppo e l'ottimizzazione delle pipeline di analisi da utilizzare per la ricostruzione di dati grezzi simulati e per la visualizzazione degli eventi in uno spazio 3D. Queste analisi sono state eseguite utilizzando `ctapipe`, un framework per la prototipazione degli algoritmi di elaborazione dati di basso livello per CTA. La struttura della libreria è presentata nel capitolo 5 insieme ad un focus sui metodi di ricostruzione che sono implementati in `ctapipe`, incluso `ImPACT`. Questo metodo utilizza una libreria di immagini, create dalle simulazioni di Monte Carlo, e un seed ottenuto

dal metodo di ricostruzione standard per trovare una migliore stima dei parametri dello sciame atmosferico. Il profiling del tempo di esecuzione e le strategie adottate per ottimizzare la pipeline ImPACT sono presentati nel capitolo 6. Viene presentata anche l'implementazione della pipeline per l'analisi del Large Size Telescope per l'osservazione in modalità monoscopica e la sua implementazione su GPU con PyTorch. ctapipe è stato anche utilizzato e sviluppato per stimare le prestazioni del CTA quando si osserva utilizzando la modalità di "puntamento divergente", in cui le direzioni di puntamento dei telescopi sono leggermente diverse rispetto alla modalità di puntamento in parallelo, in modo che l'iper-campo visivo finale di tutti i telescopi sia più ampio rispetto alla modalità di puntamento in parallelo. Le risoluzioni angolari ed energetiche e anche la sensibilità sono peggiori in questo scenario, ma avere un ipercampo di vista più ampio può essere utile per alcuni obiettivi scientifici, come la ricerca di fonti transienti. Le modifiche al codice di ricostruzione introdotte in ctapipe e i grafici relativi alla risoluzione angolare per sorgenti puntiformi sono presentate nel capitolo 7. Il capitolo finale presenta le conclusioni relative al lavoro svolto.

I risultati presentati in questa tesi sono una dimostrazione dell'uso di tecniche software avanzate in astrofisica ad altissima energia.

Abstract

All the work in this thesis has been developed in the context of the Cherenkov Telescope Array (CTA), which is going to be the major next-generation observatory for ground-based very-high-energy gamma-ray astronomy. The plan for this work is to use GPUs and Cloud Computing in order to speed up the computing demanding tasks, developing and optimizing data analysis pipelines.

The thesis consists on two main parts: the first one is dedicated to the estimation of the future performances of CTA towards the observation of violent phenomena such as those generating Gamma Ray Bursts and Gravitational Waves, with a initial work done for the creation of the models for the First CTA Data Challenge. The second part of the thesis is related to the development of the pipelines for the reconstruction of the low-level data coming from the Monte Carlo simulations using the software library called ctapipe. In chapter 1 I go into the details of the CTA project, the telescopes and the performances of the array, together with the methods used to derive them from Monte Carlo simulations. The science goals of CTA and the Key Science Projects (KSPs) will be covered in chapter 2, with a focus on Gamma Ray Bursts and the follow-up of Gravitational Waves events.

The work done for the First CTA Data Challenge (DC-1) is presented in chapter 3. More than 500 extragalactic sources have been modelled combining informations from different catalogues in order to create a population of AGNs. This Challenge has been important both to involve more people in the analysis of CTA data and to compute the observation time needed by the different KSP. The simulations for the gravitational waves and gamma-ray bursts Consortium papers have been created with the `ctools_pipe` pipeline (presented in chapter 4), implemented around the libraries `ctools` and `gammalib`. The pipeline is composed of two main parts: the task to be executed (background simulation, model creation and detection) and in which computing centre.

The second part of the thesis is focused on the development and the optimization of the analysis pipelines to be used for the event reconstruction of simulated raw data and for the visualization of the events in a 3D space. This analyses have been performed using ctapipe, a framework for prototyping the low-level data processing algorithms for CTA. The structure of the library is presented in chapter 5 together with a focus on the reconstruction methods that are implemented in ctapipe, including the so called ImPACT. This method uses a template of images created from the Monte Carlo simulations and a seed from the standard reconstruction method to fit between the templates to find a

better estimation of the shower parameters. The time profiling and the strategies adopted to optimize the ImPACT pipeline are presented in chapter 6. The implementation of the pipeline for the analysis of the Large Size Telescope observing in monoscopic mode and its GPU implementation with PyTorch is also presented. `ctapipe` has also been used and developed to estimate the performances of CTA when observing using the “divergent pointing” mode, in which the pointing directions are slightly different with respect to the parallel pointing mode, so that the final hyper field-of-view of all the telescopes is larger with respect to the parallel pointing mode. The angular and energy resolutions and also the sensitivity are worse in this scenario, but having a wider hyper field-of-view can be good for other topics, such as searching for transient sources. The modifications to the reconstruction code introduced in `ctapipe` and some angular resolution plots for the simulated point source gammas are presented in chapter 7. The final chapter presents the conclusions of the thesis.

The results presented in this thesis are a demonstration of the usage of advanced software techniques in very high energy astrophysics.

Résumé

Tous les travaux de cette thèse ont été développés dans le contexte du Cherenkov Telescope Array (CTA), qui sera le principal observatoire de la prochaine génération pour l’astronomie gamma à très haute énergie au sol. Le plan de ce travail est d’utiliser les GPU et le Cloud Computing afin d’accélérer les tâches de calcul exigeantes, en développant et en optimisant les pipelines d’analyse de données.

La thèse se compose de deux parties: la première est destinée à l’estimation des performances du CTA pour l’observation de phénomènes violents tels que ceux générant des sursauts de rayons gamma (GRB) et des ondes gravitationnelles, avec un premier travail effectué pour la création des modèles pour le premier CTA Data Challenge (DC1). La deuxième partie de la thèse est liée au développement des pipelines pour la reconstruction des données de bas niveau provenant des simulations de Monte Carlo.

Dans le chapitre 1, je présente les détails du projet CTA, les télescopes et les performances du réseau, ainsi que les méthodes utilisées pour les dériver des simulations de Monte Carlo. Les objectifs scientifiques du CTA et les Key Science Projects (KSP) seront couverts dans le chapitre 2, avec un accent sur les GRB et le suivi des événements liés aux ondes gravitationnelles.

Plus de 500 AGNs ont été modélisées pour le DC1, qui a été important à la fois pour impliquer davantage de personnes dans l’analyse des données du CTA et pour calculer le temps d’observation nécessaire aux différents KSP. Les simulations pour les papiers du Consortium sur les ondes gravitationnelles et les sursauts gamma ont été créés avec le pipeline `ctools_pipe` (présenté au chapitre 4), mis en œuvre autour des bibliothèques `ctools` et `gammalib`. Le pipeline est composé de deux parties: la tâche à exécuter (simulation de fond, création de modèle et la partie qui effectue la détection) et dans quel centre de calcul.

La deuxième partie de la thèse est axée sur le développement et l’optimisation des pipelines d’analyse à utiliser pour la reconstruction d’événements à partir de données brutes simulées et pour la visualisation des événements dans un espace 3D. Ces analyses ont été réalisées à l’aide de `ctapipe`, un framework pour le prototypage des algorithmes de traitement de données de bas niveau pour CTA. La structure de la bibliothèque est présentée dans le chapitre 5, avec un accent particulier sur les méthodes de reconstruction qui sont mises en œuvre dans `ctapipe`, y compris le système `ImPACT`. Cette méthode utilise un modèle d’images créé à partir des simulations de Monte Carlo et une “seed” de

la méthode de reconstruction standard pour s'adapter entre les modèles afin de trouver une meilleure estimation des paramètres de la gerbe atmosphérique. Le profilage temporel et les stratégies adoptées pour optimiser le pipeline ImPACT sont présentés au chapitre 6. L'implémentation d'un pipeline pour l'analyse de l'observation du Large Size Telescope en mode monoscopique et son implémentation GPU avec PyTorch est également présentée. ctapipe a également été utilisé et développé pour estimer les performances du CTA lors d'observations en mode "pointage divergent", dans lequel les directions de pointage sont légèrement différentes par rapport au mode de pointage parallèle, de sorte que l'hyper champ de vision final de tous les télescopes est plus grand par rapport au mode de pointage parallèle. Les résolutions angulaires et énergétiques ainsi que la sensibilité sont moins bonnes dans ce scénario, mais le fait d'avoir un hyper field-of-view plus large peut être bénéfique pour d'autres sujets, comme la recherche de sources transitoires. Les modifications du code de reconstruction introduites dans ctapipe et certains tracés de résolution angulaire pour les gammas de source ponctuelle simulés sont présentés au chapitre 7.

Les résultats présentés dans cette thèse sont une démonstration de l'utilisation de techniques logicielles avancées en astrophysique de très haute énergie.

Chapter 1

The Cherenkov Telescope Array

In this chapter I will describe the working principles of the Imaging Cherenkov telescopes and show the existing arrays of IACTs. I'll then go into the details of the Cherenkov Telescope Array, the sites in which the telescopes will be built and the telescope types. All the characteristics of the instruments are derived from the science goals that are expected to be achieved with CTA.

After the description of the various telescope types, I'll show the results that have been obtained from the latest Monte Carlo production showing the data produced, the software used to get the performances of CTA and the data analysis pipeline used for the creation of the Instrument Response Functions.

1.1 Imaging Cherenkov telescopes

The electromagnetic spectrum ranges from the radio waves to the TeV gamma rays, but only radio waves above a certain frequency, some infrared wavelengths, visible light and ultraviolet light are able to reach the ground. High energy photons can be observed directly only with a satellite, such as *Fermi*, which detects gamma rays from their interaction with the satellite and the conversion of such particle in an electron-positron pair, which are then tracked across the silicon layers down to the calorimeter which measures the energy of the pair.

Due to the low flux of particles at the highest energies, the collection area offered by satellites for the observation of Very High Energy (VHE) photons (above 100 GeV) is limited. The observation of these photons can be done in an indirect way from the ground using Imaging Atmospheric Cherenkov Telescopes (IACT). These are optical-UV telescopes observing the Cherenkov radiation which is emitted by the medium that has been perturbed by secondary particles created by a primary VHE particle while traveling through the atmosphere. The development of the particle shower in the atmosphere depends on the nature of the primary particle and while gamma rays and electrons are producing electromagnetic showers, protons or other hadrons produce hadronic showers (see figure 1.1). The difference in the development of an electromagnetic and an hadronic showers can be seen in figure 1.2, with images obtained from Monte Carlo simulations.

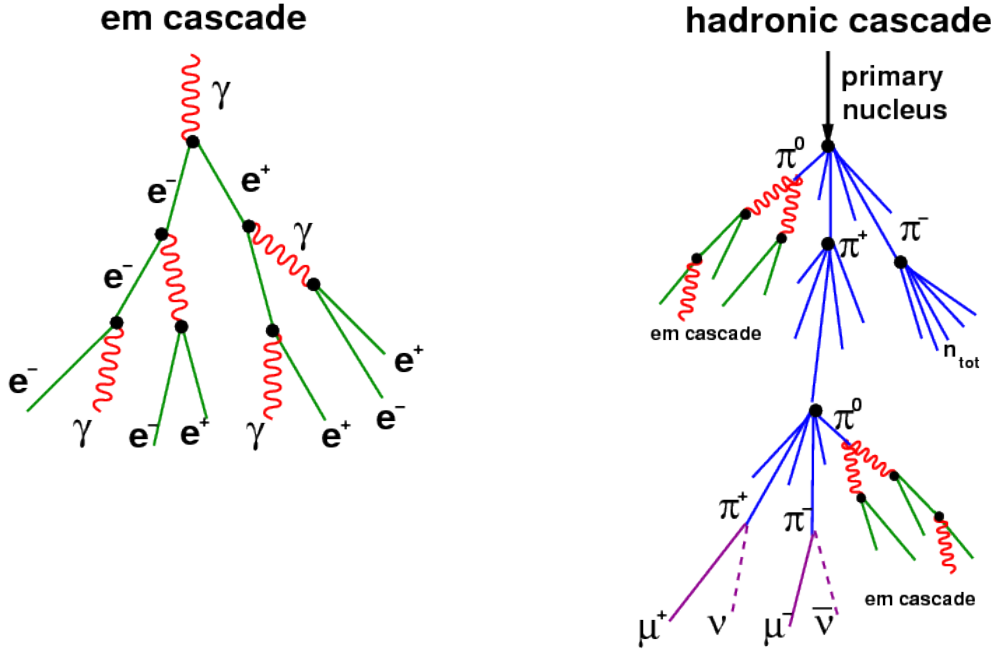


Figure 1.1: development of an electromagnetic shower (left) and an hadronic one (right) Figure from [1].

When a gamma ray enters in the Earth's atmosphere, it interacts with the molecules in air at an height of roughly 20–30 km above sea level and it is converted in an electron-positron pair. These particles initiate an electromagnetic shower producing new secondary gamma rays via bremsstrahlung which are then converted into an electron-positron pair, and so on. This process becomes subdominant with respect to ionization losses when the energy of the secondary particles goes below a critical energy E_c , which is the energy for which the energy losses by bremsstrahlung and ionization are equal (the E_c in air is 86 MeV). An important parameter to take into account is the height at which the number of particles produced by the electromagnetic shower is maximum, which is known as height of the shower maximum, or h_{max} . We can also estimate this height considering the depth from the top of the atmosphere at which this maximum emission is reached, which is measured in units of radiation length and is often referred to as x_{max} : the higher this value, the lower is the height of the shower maximum. This parameter is connected with the energy of the particle, with a more energetic particle reaching a lower h_{max} (a larger x_{max}) and viceversa.

The development of a shower initiated by a hadron is different with respect to the one initiate by a gamma ray because the first interaction of this particle with the air molecules is governed by the strong force and particles such as the pions are produced. Pions also decay into photons, electrons, positrons and muons generating secondary electromagnetic showers or muon-generated showers. Due to the nature of the interaction, the transverse momentum of secondary hadrons is larger than that of secondary leptons in the electromagnetic shower, resulting in hadronic showers being wider.

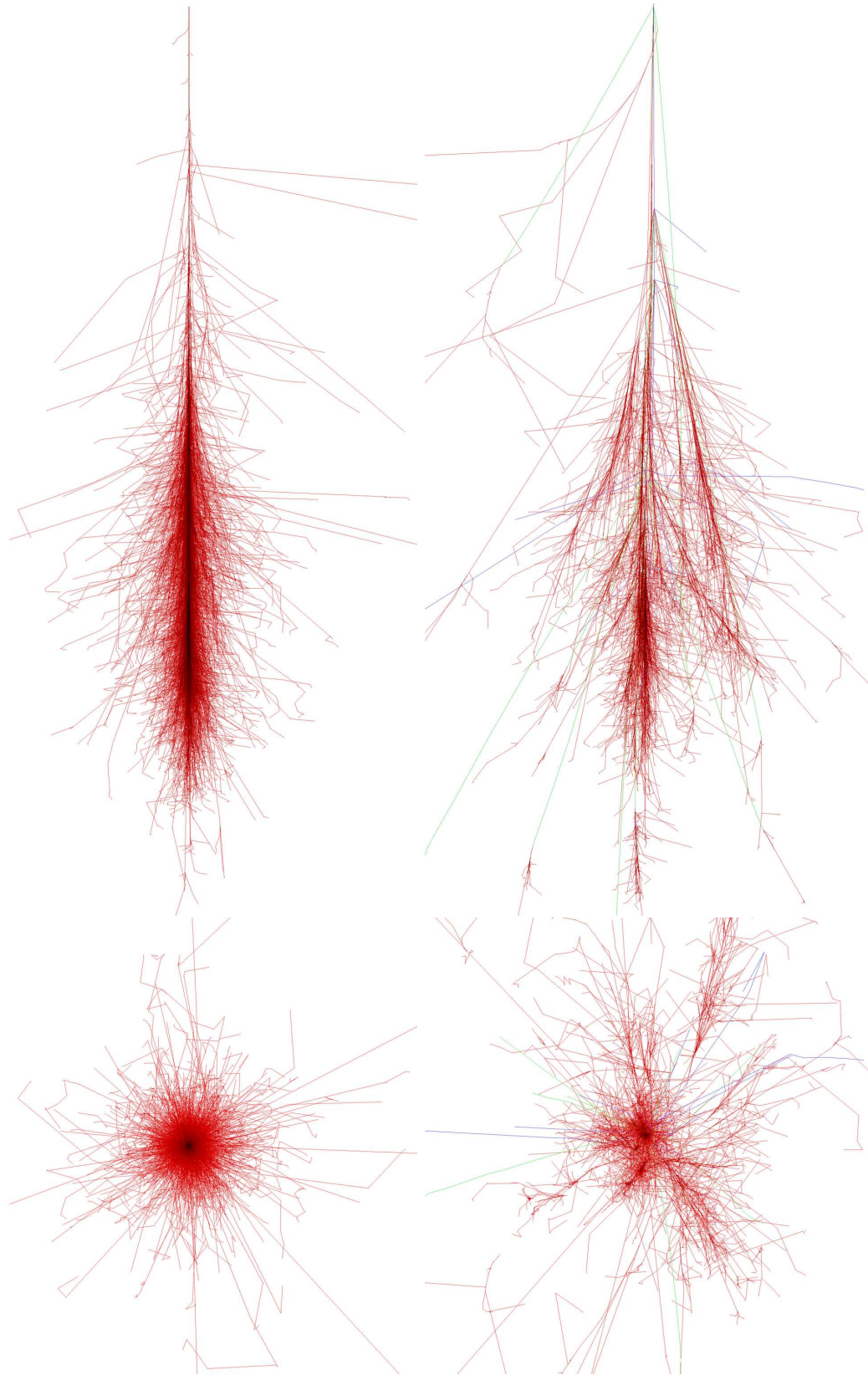


Figure 1.2: particle tracks for a photon (left) and a proton (right) with initial energies of 100 GeV, simulated using CORSIKA (<https://www.ikp.kit.edu/corsika/>). The upper panels reports the longitudinal projections and the lower panel is reported in the lower panel. Red tracks are e^- , e^+ and gamma rays, green are muons and blue are hadrons.

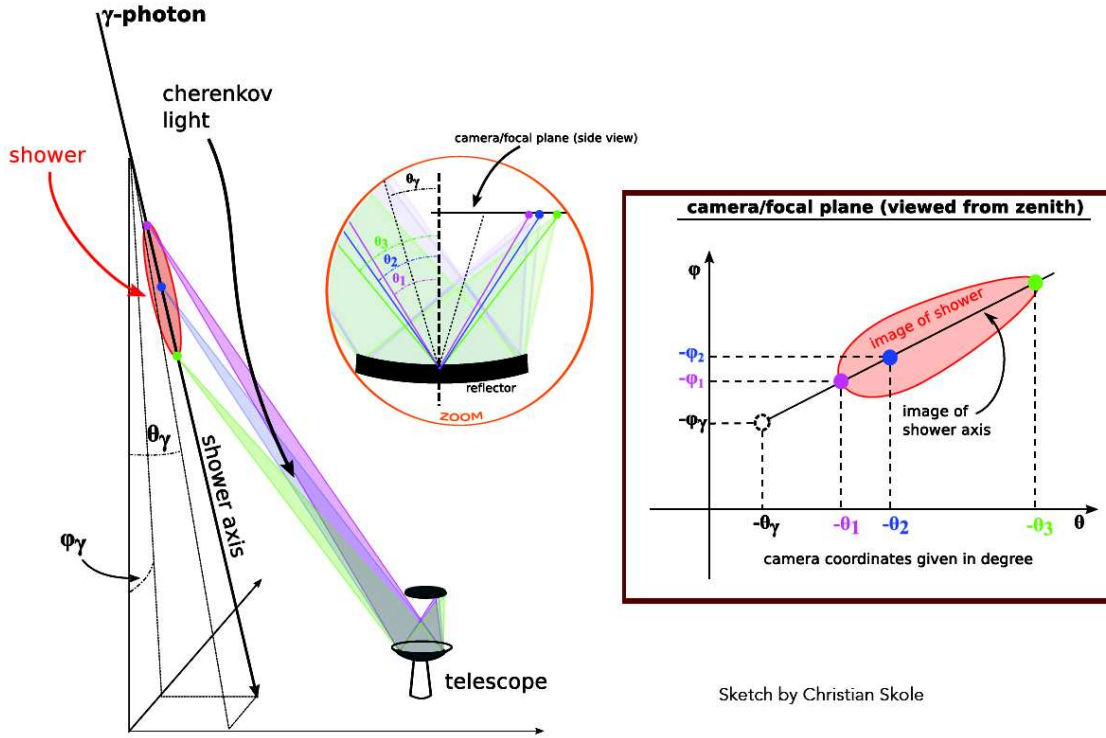


Figure 1.4: sketch showing where the Cherenkov photons produced in the atmosphere are collected in the camera. For a telescope close to the impact point of the shower, the light from the green point reaches the camera before the light from the blue and purple points (credits to C. Skole and G. Maier).

Cherenkov emission Due to the high energy of the primary particle, the secondary charged particles, such as electrons, positrons and pions, are produced with a high energy such that they travel in air with a speed v greater than the speed of light in that medium, c/n . This causes these particles to interact with the medium which then reacts emitting a radiation in the UV-optical region called Cherenkov radiation. The radiation is emitted in the form of a cone at an angle θ such that:

$$\cos \theta = \frac{c}{vn(\lambda)} \quad (1.1)$$

where $n(\lambda)$ is the spectral index of the medium, which depends on the wavelength of the Cherenkov light emitted. Since the refractive index of air is very close to one, $(n - 1) \sim \mathcal{O}(10^{-4})$, and the particle velocity is close to the speed of light, the angle of the Cherenkov cone is smaller than 1° (see figure 1.3).

The Cherenkov radiation produced by the particle shower is very collimated, with an

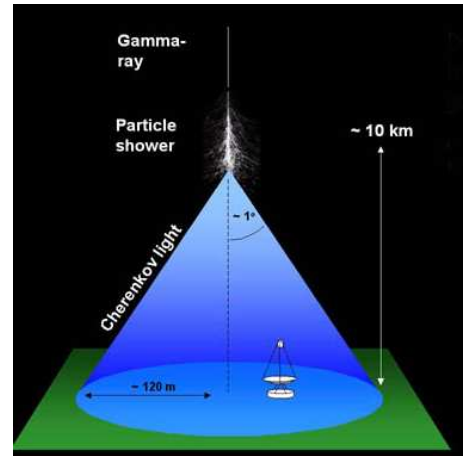


Figure 1.3: development of atmospheric particle shower (from H.E.S.S. website <https://www.mpi-hd.mpg.de/hfm/HESS/>).

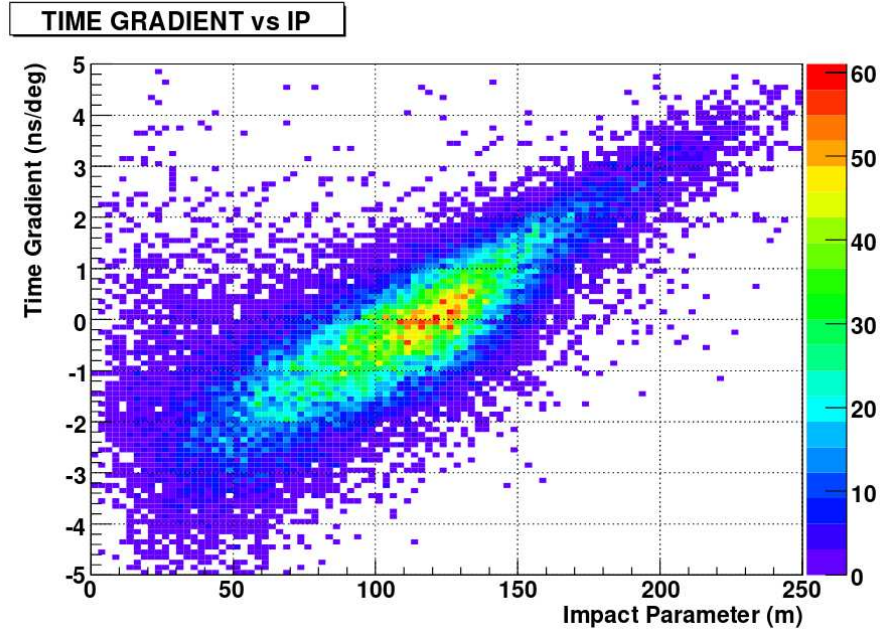


Figure 1.5: scatter plot created from Monte Carlo simulations of gamma ray events showing the time gradient (ns/deg) vs the distance of the impact point from the telescope axis. Figure from [3].

angle θ depending on the index of refraction of the air which varies with the altitude. A vertical incident gamma ray produces on the ground a circular light pool with a radius of roughly 120 meters. The resulting time-integrated image in the camera, cleaned from the noise signal, is a projection of the 3D development of the shower (see figure 1.4) and the moments of such image are used for the reconstruction of the parameters of the primary particle. The parameters of the ellipse, called Hillas parameters [2], are the position of the maximum emission in the camera and its distance and angle with respect to the centre of the camera, the width, the length, the skewness and the rotation angle with respect to the camera x-axis (see also chapter 5).

The development of the shower in the atmosphere is really fast and it is observed as a nanoseconds long burst. Moreover, due to the difference between the velocity of propagation of the particles in the shower and the Cherenkov light produced, there is a delay between the arrival time of the Cherenkov photons produced at the different heights. The value of this delay depends on the position between the position of the impact point of the particle on the ground and the position of the telescope. If the impact point is very close to the telescope position (less than ~ 100 m), then the first light which will be observed will be the one coming from the bottom of the shower (green light in figure 1.4), otherwise if the telescope is far from the impact point the upper part of the shower (purple light in figure 1.4) will arrive on the telescope before the lower part (see figure 1.5). The time information of the development of the shower in the cameras of the telescopes is a useful parameter for the analysis of the images in the camera and for the reconstruction of the shower parameters.



(a) VERITAS.



(b) MAGIC.



(c) H.E.S.S.

Figure 1.6: the three existing arrays of IACTs (MAGIC image from https://www.mpg.de/5571159/The_MAGIC_Telescopes, H.E.S.S. image from <https://www.mpi-hd.mpg.de/hfm/HESS/>) and VERITAS image from <http://veritas.sao.arizona.edu/>.

1.1.1 IACTs

Since the beginning of the last decade, many IACT arrays have been built around the world and the three biggest arrays of IACTs, are MAGIC, VERITAS and H.E.S.S. and their performances together with those of the two CTA sites is reported in figure 1.8. Other single telescope IACT experiments are FACT (First G-APD Cherenkov Telescope), built at the Roque de Los Muchachos in La Palma, which uses a camera with pixels made of Geiger-mode avalanche photodiodes, and the Major Atmospheric Cherenkov Experiment (MACE), built in India at an height of 4270 meters above sea level.

MAGIC: the Major Atmospheric Gamma-ray Imaging Cherenkov Telescope, located in La Palma (Canary Islands, Spain) and operating since 2004, is made of two telescopes with a diameter of 17 m each. There has been only one telescope until 2009 and then a second one was added, so that since then the system has been working in a stereo configuration. MAGIC covers the energy range from 30 GeV to 100 TeV and the telescopes can repoint to any point in the sky within 30 seconds.

H.E.S.S.: the High Energy Stereoscopic System, is an array of four IACT with a diameter of 12 m, organized in a square configuration of 120 m side and operational from 2004 in Namibia, that have been upgraded in 2012 with an additional bigger telescopes of 28 m of diameter put in the center of this square configuration. This goal of this bigger telescope is to decrease the low energy threshold and increase the sensitivity at low energies.

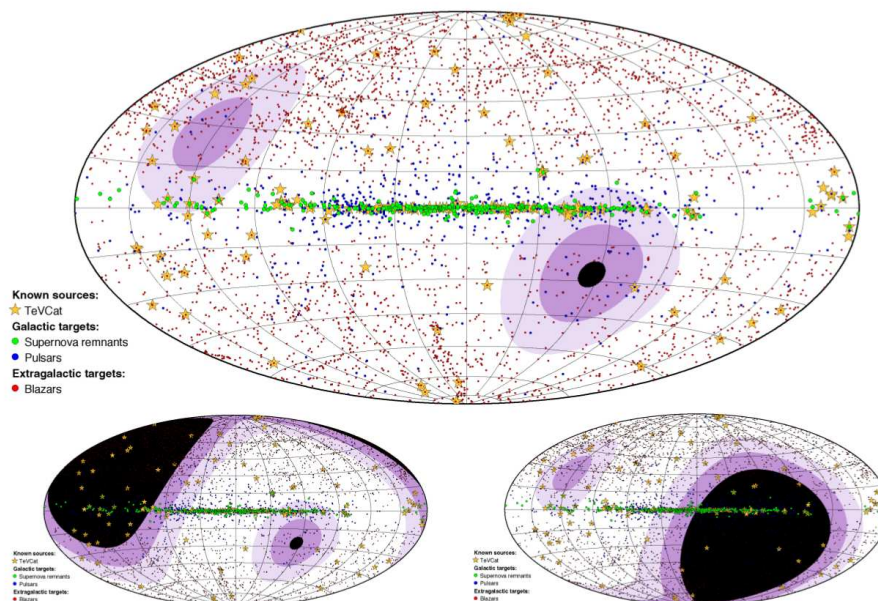


Figure 1.7: Sky coverage obtained with the two CTA array sites, compared to the Southern (bottom left) or Northern (bottom right) observatory alone. The sky is shown in Galactic coordinates, with the Galactic plane along the equator. Indicated are VHE gamma-ray sources, and blazars, supernova remnants (SNRs) and pulsars as key target classes. The color scale indicates the minimum zenith angle under which a target is visible, from 0° – 30° (white) to 30° – 45° , 45° – 60° and $> 60^\circ$ (black). The Cherenkov technique works best at small zenith angles ($< 30^\circ$); 60° is the practical limit. Figure from [4].

VERITAS: the Very Energetic Radiation Imaging Telescope Array System is located at the Fred Lawrence Whipple Observatory in southern Arizona, USA. It’s an array of four IACTs with primary mirrors with a diameter of 12 m, a field of view of about 3.5° and capable of observations in the 50 GeV – 50 TeV energy range.

1.2 CTA sites and telescopes

The creation of Monte Carlo simulations is a really important process needed to estimate the performances of CTA, guide the design of the different telescope types and compare their capabilities. These simulations provide input to the site selection process by evaluating the effect of the characteristics of each site on the performances of the whole array. The simulations are analysed to extract the performances of a particular configuration on the observation of a source. In this process there are many parameters that are needed as input, e.g., the modelisation of the atmosphere of the possible site, the field of view of the telescope or the response of the photomultipliers in the camera.

In order to choose the site in which the telescopes will be built and choose the optimal layout, three Monte Carlo productions have been produced and analysed in the past years: those productions led to the selection of the sites and to the determination of the number of telescopes per telescope type in each site (see figure 1.8).

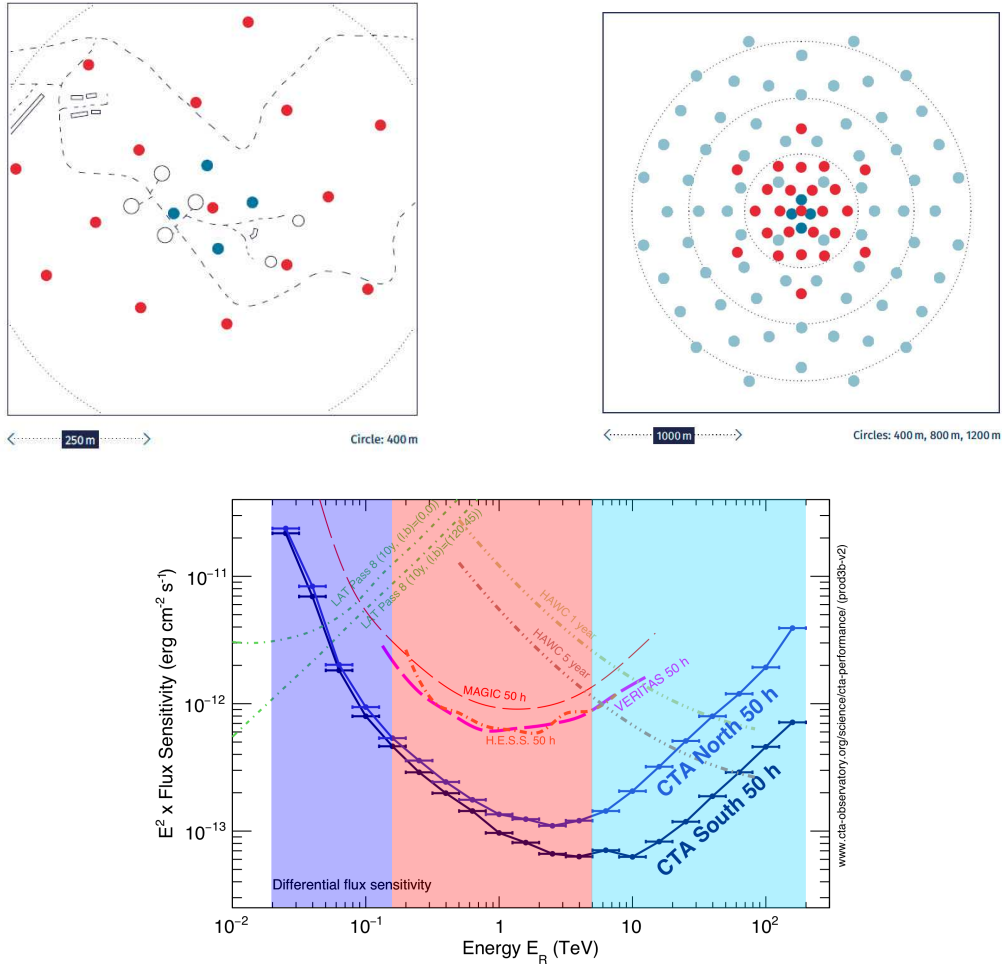


Figure 1.8: *Top panel:* the figure on the left represents the telescopes layout of the North site in La Palma, with 4 LSTs and 15 MSTs. The figure on the right is the South site in Cerro Paranal with 4 LSTs, 25 MSTs and 70 SSTs. *Bottom panel:* sensitivity curves for both arrays, together with those of existing instruments, with the highlight of the energy range at which the telescopes subarrays provide full system sensitivity (LSTs, MSTs and SSTs).

The aim of the first two Monte Carlo productions (named **prod1** and **prod2**) was the site selection and the exploration of very different layouts, from those focused on low energies, to very extended ones, focused on multi-TeV energies: those production showed that the sites at a moderate altitude of roughly 2000 m a.s.l. would give the best overall performances [5]. The two selected sites are Cerro Paranal (Chile) and Roque de Los Muchachos in La Palma (Canary Islands, Spain): the selection of these sites have a direct impact on the observability of some object in the sky (as it can be seen in figure 1.7).

The **prod3** was carried out for the two selected sites, using updated and more realistic telescopes models with the goal to find the optimal telescope positions which maximizes the overall performances of CTA.

After the analysis of **prod3** and **prod3b**,¹ the optimal final layout for the two sites,

¹The **prod3b** for Paranal stretches the LST position, introduces an additional MST in the center and smooths the position of the SSTs; the changes for La Palma are mainly coming from topography constraints.

together with number of telescopes per each size in each site, was found to be the following:

North site: 4 Large Size Telescopes and 15 Medium Size Telescopes;

South site: 4 Large Size Telescopes, 25 Medium Size Telescopes and 70 Small Size Telescopes.

Along with the three telescope sizes, there are many sub-divisions since many different groups in the Consortium are working on different cameras and telescope prototypes. While for the LST there is just one camera (the “LSTCam”) and one telescope structure proposed, the proposed MST structures are a modified Davies-Cotton configuration, and a Schwarzschild-Couder optical configuration; the cameras proposed for the MST are the “FlashCam” and the “NectarCam” for the DC configuration and the “SCTCam” for the SCT configuration. While NectarCAM will use the “Nectar” analog pipeline ASIC for signal capture, the readout of the FlashCam is fully digital and both cameras are equipped with photomultiplier tubes (PMTs) with arranged in an honeycomb configuration. The SCTCam has more than 11000 silicon PMTs (SiPM) arranged in a squared configuration. For what concerns the SSTs, there are three different proposed telescopes; while the ASTRI prototype and the GCT prototype have a Schwarzschild-Couder configuration, the SST-1M has a Davies-Cotton optical configuration. These telescopes will be equipped with “ASTRI-Cam”, “CHEC-Cam” and “DigiCam” respectively. While these are the telescope and camera types that have been used in the **prod3** Monte Carlo simulations, the SST are undergoing a “remodelling” process to reach a single design between the proposed ones (see SST section later).

In the following sections I’ll present the different telescopes and cameras highlighting some key characteristics and capabilities that make their construction really necessary for the full accomplishment of CTA scientific goals.

1.2.1 Large Size Telescope

Gamma rays with energies below few hundreds of GeV produce a small amount of Cherenkov light, so that very-large-aperture telescopes are needed in order to collect this light. Moreover, since the background is higher at those energies, the sensitivity is limited by the background systematics and therefore we need an array of those large aperture telescopes to be arranged in a compact array. This maximizes the light collection area and reduces the background coming from light produced by secondary muons and, with further analysis, also the background coming from protons.

The Large Size Telescopes (LSTs) will be dominant in the low energy range, from 20 GeV up to 200 GeV, and there will be 4 LSTs in each array, placed at the core of the array with an average inter-telescope separation of ~ 150 meters; they will be the biggest telescopes to be built in the CTA array, with a primary reflector of 23 m in diameter.

The telescope has an altitude-azimuth mount with a tubular structure made of reinforced carbon fiber tubes and steel tubes: the main parts of the structure are the "U"-

shaped mount which has all the motors necessary to move the telescope, the dish which holds all the 196 mirrors, and the Camera Support Structure (CSS) with the camera frame at its vertex hosting the camera itself. The structure of the LST is very similar to the one of the MAGIC telescopes, hosted in the same site of CTA North.



Figure 1.9: Prototype (status on the 27/9/2018) of the first LST being built in La Palma (from CTA Flickr webpage, Credit: Victor Acciari).

The camera is made of 1855 photomultiplier tubes equipped with optical light concentrators, grouped in 265 modules with 7 PMTs each, with the readout electronics based on the DRS4 chip, already used by the MAGIC telescopes. The camera has high-gain and low-gain channels in order to extend the dynamic range of the signal that can be recorded. The field of view of the camera is about 4.5 degrees.

Its low energy threshold, the lowest ever achieved by an array of IACTs, together with its fast re-positioning capability, less than 30 seconds to point towards any direction in the sky, will allow the study of transient phenomena. Gamma-ray bursts at a really low energy will be observed, in order to both perform the detection of such transient sources and obtain high-quality spectra, which will provide informations on the emission mechanism happening in those sources. The re-positioning speed and the low energy threshold provided by the LSTs will be necessary in order to study transient source, from the galactic ones to those at high-redshift such as Active Galactic Nuclei and GRBs.

1.2.2 Medium Size Telescope

In the middle-energy range, from hundreds of GeV up to tens of TeV, the Cherenkov light produced is greater and therefore the aperture can be reduced. The sensitivity is improved by increasing the number of telescopes, with a limit basically imposed by the cost per telescope.

The Medium Size Telescopes will dominate the CTA sensitivity in the energy range between 150 GeV up to 5 TeV, with a foreseen number of 15 telescopes to be built in La Palma and 25 MST in Chile, and an average inter-telescope separation larger with respect to the LSTs in order to increase the total light collection area. While the modified Davies-Cotton will have just a primary mirror with a diameter of 11.5 m, the Schwarzschild-Couder configuration will have a primary mirror of 9.7 m and a secondary mirror of 5.4 m, with a field-of-view ranging from 7.5° to 7.7° depending on the camera type. The SCT is very different when compared to the other MSTs not only for what concerns the optical design



Figure 1.10: *Left figure:* Prototype of the MST in Desy, Zeuthen (Berlin) with the NectarCam installed (from CTA Flickr webpage). *Right figure:* prototype of the SCT telescope at the Whipple Observatory in Arizona (Credit: Amy Oliver).

but also for what concerns the camera: while the NectarCam and FlashCam will have 1855 and 1764 PMTs respectively, the SCT will have 11328 Silicon photomultipliers (SiPM). This double-mirror configuration and the small pixel size allows to better focus the light.

All the proposed MSTs configurations will be able to point to any point in the sky within 90 seconds, which is greater with respect to the LSTs repoint time, making them less prone to the prompt detection of transient sources.

Due to their large number and their large field-of-view, the MSTs will be really important to perform blind surveys outside the Galactic plane looking for flares in extragalactic sources and to perform a monitoring of those sources known to be rapidly variable.

Due to their large field-of-view and the huge number of MST that will be built, together with their sensitivity in the mid-energy domain, those telescopes will be the perfect instrument to perform the sky survey in the so-called “divergent pointing mode”. While in the standard parallel pointing mode the field-of-view of the telescopes projected in the sky overlap completely, when pointing in divergent mode they only partially overlap, creating a hyper field-of-view with a lower average multiplicity with respect to the parallel pointing. The effect of this pointing mode is to increase the total field-of-view of the subarray of telescopes which is observing a certain patch of sky, at the expense of a lower angular and energy resolution.

The analysis done in the divergent pointing mode will be covered in section 7.1.

1.2.3 Small Size Telescope

The flux of extragalactic gamma rays at the highest energies (above ~ 1 TeV) is very small and it’s also absorbed due the interaction with the EBL, the most energetic gamma



Figure 1.11: 3D models of the SSTs. The first two models use the Schwarzschild-Couder double mirror configuration while the SST-1M uses only a primary mirror.

rays that CTA will detect will come from our Galaxy. Since CTA South will have a much better view of the Galactic plane compared to CTA North, there will be no SSTs in La Palma but only in Chile (see figure 1.7).

The high-energy domain that CTA is going to cover (from few TeV up to 300 TeV), is a region of the electromagnetic spectrum in which there is less background and there is a huge amount of Cherenkov photons produced at by primary gamma rays: the telescopes observing at these energies can be smaller with respect to the MSTs, allowing for a lower cost for each telescope. The only limitations at these energies are coming from the low flux of gamma rays and in order to increase the light collection area, it's necessary to have many telescopes spread over a large area. There will be 70 SSTs spread over 4 km² with a inter-telescope distance ranging from 300 m for the inner ones to more than 500 m for the outer ones in the array.

CTAO project manager has recently published a note² in which it is stated the outcome of the SST harmonization process in which only one type of SST will be built in Chile between the three proposed ones: having only one telescope type will make the operation of maintenance easier during the lifetime of the array.

In order to choose between the SST types, or between another prototype which might be a combination of camera and telescope structure, a new Monte Carlo production (**prod4**) was needed. In this production the single prototypes were simulated, together with many different mixed configurations of telescopes and cameras. The final configuration for the final CTA-SST design will be based on the ASTRI telescope for the structure and on the CHEC camera.

Three different proposed prototypes for the SSTs have been simulated in the **prod3**: the ASTRI and the GCT prototypes have a Schwarzschild-Couder configuration with primary mirrors with a diameter of 4.3 m and 4.0 m respectively and secondary mirrors of 1.8 m and 2.0 m respectively, while the SST-1M prototype have a primary mirror with a diameter

²<https://www.cta-observatory.org/small-sized-telescope-harmonization-process-and-status/>

of 4.0 m. All the cameras are equipped with SiPM in a number varying from 1296 for SST-1M, to 2048 for the GCT to 2368 for ASTRI, with their field-of-view being 8.8°, 8.3° and 10.5° respectively.

1.3 CTA Performances

The estimation of the performances of CTA goes through several steps, from the Monte Carlo simulation of the air shower generated by the incoming particle, to the propagation of simulated Cherenkov photons through the telescope structure up to the camera, with different particle types being simulated. The data from the simulation of each particle type are analysed and combined to get high-level Instrument Response Functions (IRFs) for each particular layout, pointing mode and observation duration.

The IRFs contain all the relevant parameters and response tables which describe the behaviour of the instrument under the observing conditions assumed at the simulation and analysis level. There is a different IRF per set of azimuth angle, zenith angle, subarray of telescopes selected, level of Night Sky Background and they are optimized for a certain duration of the observation. When the telescopes will be operating, each observation run will be accompanied by its IRF.

Each IRFs contains the effective area, point spread function, energy dispersion and background rate as a function of the energy and for increasing angular distances from the centre of the field of view:

Effective area: the effective area, which is a measure of the collection area of the telescope system, is lower at low energies then it increases with increasing energy and reaches a plateau at the highest energies. The effective area also degrades with increasing off-axis angle (see **prod3b** results in the next section).

Energy dispersion: the energy dispersion is the ratio between the reconstructed energy E_R and the true energy E_T from the MC simulation, and this value should be as close as possible to 1, meaning that the energy has been reconstructed properly. The energy dispersion is usually represented with the width (energy resolution) and the average (energy bias) of the $(E_T - E_R)/E_T$ distribution, with the full energy migration matrix provided in each IRFs.

Point spread function: the point spread function describes the response of the detector to a point source, which cannot be seen as a perfect point source due to interaction with the medium and the instrument, but it usually has a certain shape (the simple case is a gaussian shape). This is derived from the Monte Carlo simulation but it can be also estimated from the observation of point sources, such as AGNs.

Background rate: this shows the (post-analysis) residual cosmic-ray background rate per square degree, plotted against the reconstructed gamma-ray energy.

The last Monte Carlo production, namely the **prod3b**, was done in order to fully determine the optimal array layout using the updated parameters for the telescopes models: this MC campaign used 2 PB of disk space and about 125 MHS06³ hours [6].

1.3.1 Results from prod3b

One of the most important outcome of the analysis is the sensitivity curve, which is the minimum flux needed by CTA to obtain a 5-standard-deviation detection of a point-like source (reported in figure 1.8). Other performance plots obtained from the Monte Carlo productions for the baseline layouts (the telescopes' positions are shown in figure 1.8) are the effective area, the angular resolution and energy resolution (see figure 1.12). The angular resolution curve reported against the reconstructed energy shows the angle within which 68% of reconstructed gamma rays fall, relative to their true direction. Gamma-hadron separation cuts are also usually applied for the MC events used to determine the angular resolution. The energy resolution is obtained from the distribution of $(E_R - E_T)/E_T$, with E_T being the true energy of the event and E_R being the reconstructed energy.

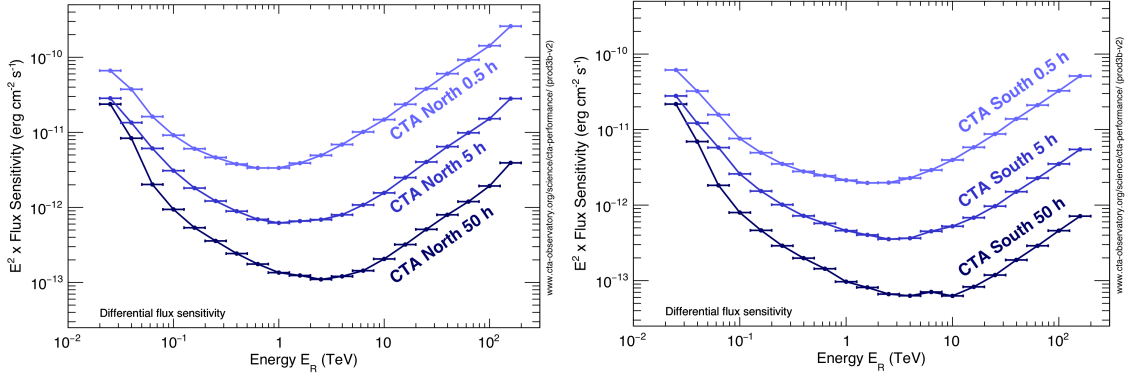
The IRFs are public and can be used by any person even though not part of the CTA Consortium, together with the high-level analysis software that are being developed for the analysis of the scientific data from CTA, such as `ctools` and `gammapy` (see section 3.2.1). The public IRFs have been created for the observation with the full array, for both the North and the South site, for 20°, 40° and 60° degrees of zenith angle, for a pointing directions towards North (azimuth = 0°), South (azimuth = 180°) and an average position between North and South and optimized for an observation time of 0.5 hours, 5 hours and 50 hours.

Together with this public version of the IRFs, there's a private version which is accessible only by the member of the CTA Consortium. The internal IRFs have been produced for a threshold implementation of the arrays (an intermediate configuration of the sites, with less telescopes, towards the full arrays), for the various subarrays (only LSTs, only MSTs, only SSTs, LSTs and MSTs only, MSTs and SSTs only) in the full or in the threshold configuration and optimized for an additional observing time of 100 seconds. In order to investigate the effect of the Night Sky Background level, Monte Carlo simulations with a NSB 5 times and 30 times the NSB level of a dark night have been created.

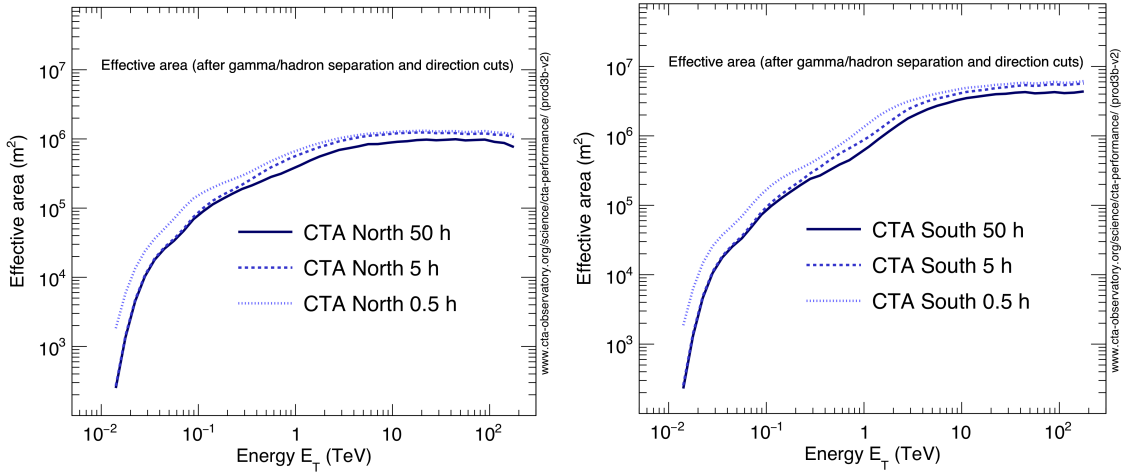
The plots presented here⁴ are those of the public IRFs reported in [7].

³HS06 is a benchmark for measuring CPU performance used in High Energy Physics with the goal of providing a consistent and reproducible CPU benchmark.

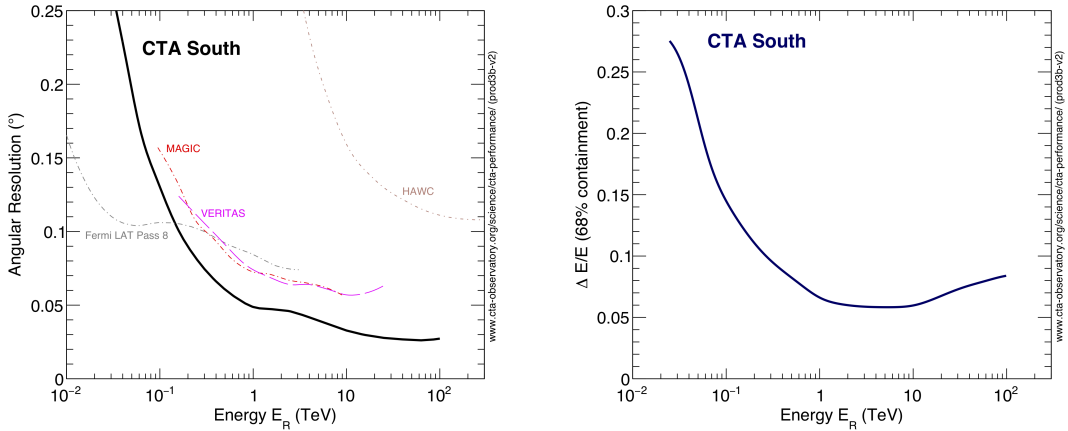
⁴There's a version on the left side of these performances plots (marked as v2, version 2, in these plots) because even though the Monte Carlo simulations is always the same, there has been small changes in the analysis method, solving some bugs that were present in the previous versions.



(a) Differential sensitivity for the North (left) and South (right) site as a function of the observing time.



(b) Effective area for the North (left) and South (right) site as a function of the observing time. This plot has been created after gamma/hadron separation and direction cuts.



(c) Angular resolution for the South (left) together with those of other experiments in the same energy range and energy resolution for the South (right), which is very similar to the one for the North site.

Figure 1.12: Some of the results from the prod3b (more in [7]). The differential sensitivity versus the observation time is reported in figure 2.12 while the differential sensitivity of CTA compared with other instruments is shown in figure 1.8.

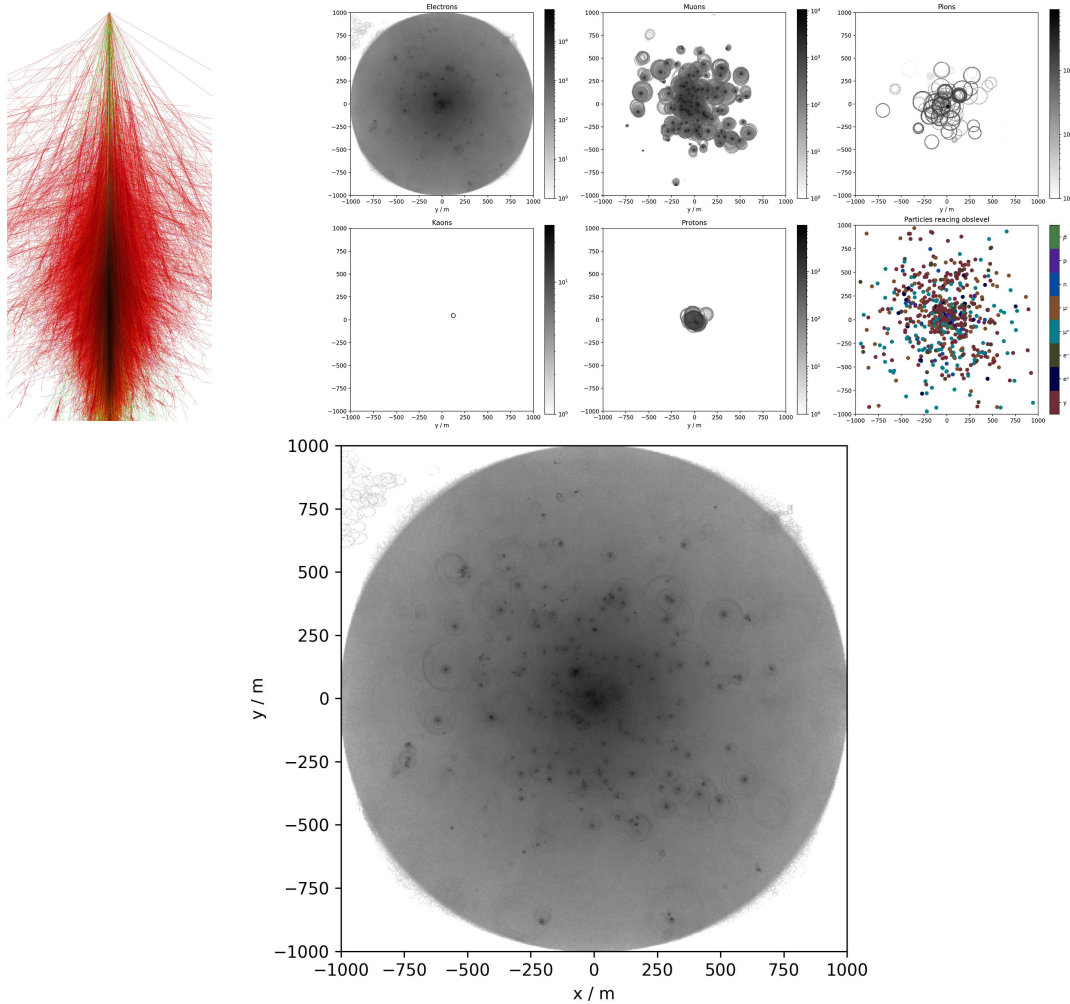


Figure 1.13: Image produced from a 10 TeV iron nuclei simulation done with CORSIKA. *Upper left:* xz-plane projection of the longitudinal development of the atmospheric shower (credit: KIT-CORSIKA). *Upper right:* Cherenkov photons reaching the ground labeled according to the particle in the cascade that originated those photons (credit: Max Nöthe). *Bottom:* all the Cherenkov photons reaching the ground (credit: Max Nöthe).

1.3.2 CORSIKA

The first step in the generation of the events seen by CTA is the simulation of the cascade of secondary particles that are generated in the atmosphere, the extensive air shower. A Monte Carlo simulation of the showers, rather than an analytical solution of the cascade equations, is necessary in order to properly take into account the shower-to-shower fluctuations.

The generation of the air showers is done with CORSIKA (COsmic Ray SIMulations for KASCADE) [8] a program for detailed simulation of extensive air showers initiated by high energy cosmic ray particles, initially built for the KASCADE experiment and then extended and used by many other experiments. Many primary particles, such as protons, photons and light nuclei up to iron can be simulated using CORSIKA. While the electromagnetic interactions are easily treated in CORSIKA, a real challenge in the simulations

is the correct treatment of the hadronic interactions, for which many different models exist and are available in CORSIKA (such as VENUS, QGSJET, DPMJET and SIBYLL, with several versions available for each model).

Many parameters can be given to CORSIKA, such as the particle type, the particle energy range, the number of particles to simulate and the spectrum to be used, the details of the site to be simulated, such as the altitude side and the strength and direction of the geomagnetic field, plus other parameters related to the simulation of the Cherenkov photons. An extension package for CORSIKA, called IACT/ATMO, is also used in order to add more functionalities related to IACT telescopes and the generation of Cherenkov light.

In order to save disk space or memory, the position of the telescopes in the ground must be provided (as x_i, y_i and z_i with respect to a ground level), so that only those photons that fall inside a sphere surrounding the telescopes positions are used in the next analysis step (see figure 1.14). Moreover, in order to save CPU time, the shower can be reused multiple times by randomly changing the impact point on the ground with respect to the original shower core.

The output can be either written to disk or piped into another program, which in the case of CTA is `sim_telarray`.

1.3.3 `sim_telarray`

A realistic and detailed simulation of the telescopes is done with `sim_telarray` [8], a software originally developed for the HEGRA experiment and later on adapted to be used by the H.E.S.S. experiment: each telescope in the array can be configured separately. With the right configuration files any other IACT system can be simulated.

The Cherenkov photons that are created from CORSIKA are ray-traced to the mirror tiles and then to the camera, where the response of the photomultipliers and the electronics is also carefully simulated. The description of the mirrors depends upon many parameters that can be also given as external files from laboratory measurements (such as the optical quality of the mirror tiles and the focal lengths of the mirror): the shadowing of camera support structure for the camera can be also taken into account. The last step in the optics simulations is to take into account the angular acceptance of the pixels, equipped with Winston cones in order to avoid gaps between neighbouring pixels as

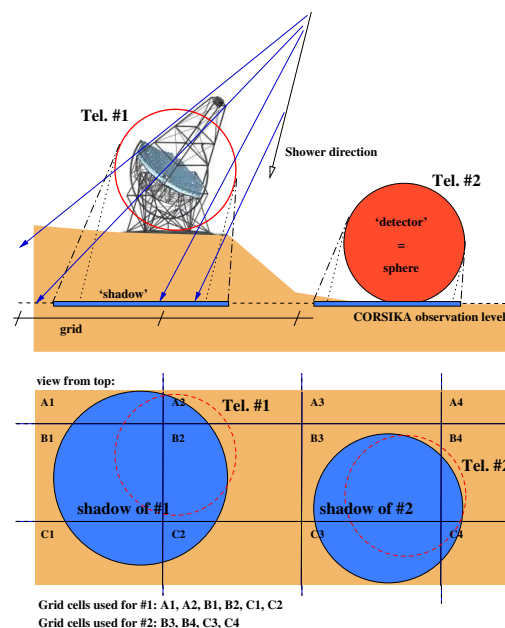


Figure 1.14: Definition of grid cells. Only those events falling in the “shadow” of the telescope are saved (or piped into `sim_telarray`) while the others are discarded. Figure from [8].

much as possible. The photon detection probability depends on many factors, such as the atmospheric transmission, the shadowing of the camera and the quantum efficiency of the photo-cathode, with most of them being functions of the wavelength. After the photons are ray-traced to the PMT pixels, the detailed simulation of the electronics is performed.

The last step in the simulation of the system is the multi-level trigger scheme: at a first stage, each telescope produces a trigger signal based on the pixel signals in the telescope camera. The final event trigger is formed by combining the trigger signals from several telescopes, with the standard requirement of having at least two telescopes triggered in order to process the event in a stereoscopic mode.

The output data of `sim_telarray` contains similar informations with respect to the raw data recorded from the telescopes. The data format for the simulated data is based on the *eventio* machine-independent format, similar to the output of CORSIKA IACT.

It was important for the work done in this thesis to understand how to change the input cards to be given to `sim_telarray` for what concerns the work done for the divergent pointing (section 7.1), since the pointing direction of each telescope had to be modified according to the pointing pattern, which is different with respect to the parallel pointing configuration that is used as default.

1.3.4 Analyses for IRF production

There are different analysis methods that have been used in order to derive the IRFs from the simulated data or to get the high-level data from the real data recorded by the telescopes.

The software packages that have been used in the past for the creation of the IRFs for CTA were originally written for other experiments and later on adapted for CTA: MARS [9] is the one used by the MAGIC telescopes, whereas EventDisplay [10] is the one used by VERITAS.

Since at the moment of writing the CTA telescopes have not obtained any data yet, a part from data from prototype telescopes, what is being done presently in order to estimate the performances of CTA, is to use the Monte Carlo simulations to produce IRFs for the arrays, and subarrays (such as the one only for LSTs or MSTs), observing at different zenith and azimuth angles. Those IRFs are then used within the science tools in order to assess the performances of a particular subarray when observing for a certain time a source with a certain spectra.

Reviewing the software used for the production of the IRFs has been an important step for understanding the work to be done with ImPACT in `ctapipe` and for the divergent pointing (see sections 5.2.2 and 7.1).

The datasets

In order to produce the IRFs, massive Monte Carlo simulations have to be produced for different particles, namely gamma rays (both “point source”, coming from one point in

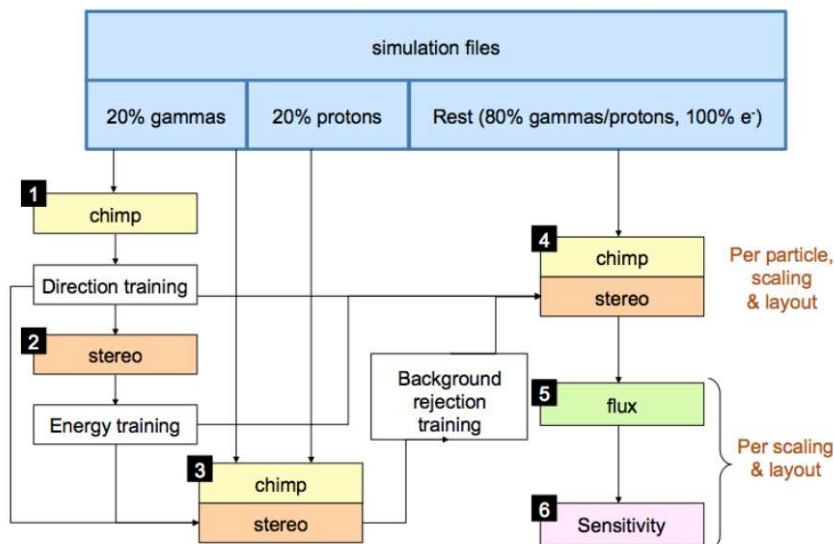


Figure 1.15: the analysis flow for MAGIC with MARS (the analysis for CTA uses different software names but the workflow is similar). Image from [6].

the sky, or “diffuse”, coming from a region of the sky), protons and electrons. In order to have a good determination of the performances of the instrument, a huge number of events have to be produced for each particle in order to properly take into account fluctuations between showers, from $2 \cdot 10^8$ for the point-source gamma rays up to $8.2 \cdot 10^9$ events for the protons.

Calibration and integration

Both analysis packages convert the output of `sim_telarray` to their own data format and proceed with the integration of the signal recorded in each pixel during the time window of data acquisition in order to get a single image.

A double-pass algorithm is used in both packages: in the first pass a sliding window with a fixed and large width is used in order to determine the time gradient of the pixels, while in the second pass a much shorter time window, centred at the time specified by the time gradient, is used to have a better charge integration (see figure 1.16).

The optimized next-neighbour (NN) cleaning is a method which considers next-neighbour groups of certain multiplicity as units for image information. The groups are selected to be pairs (2NN), triplets (3NN) or quadruplets (4NN) of next-neighbour pixels plus a 2NN+1 group which also uses the information on a second-nearest neighbour. In order to assign one group to the shower image and not discard it, the pixels in the group are required to have a charge above a certain threshold. The pulse time coincidence should also be below a certain value [11], since neighbour pixels with shower signal have a peak pulse time which has a really low spread. If a group satisfies those two conditions, then it is kept after the cleaning, otherwise it is discarded.

The two thresholds that are used in this optimized NN cleaning method are calculated

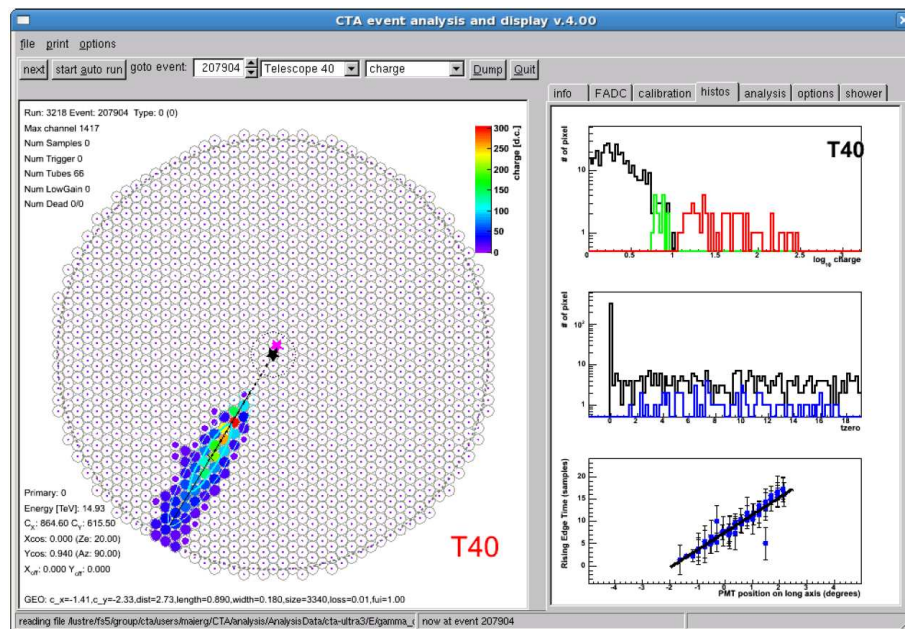


Figure 1.16: camera image for a random event analyzed with EventDisplay. The right panel shows the distribution of integrated charges in the pixels (all, image pixel and border pixel), the distribution of pulse arrival times (50% of rise time) for image and border pixels, and the time gradient along the major axis of the image used for the “double-pass” trace integration. Figure from [10].

from NSB simulations for each camera type and multiplicity group and just one parameter have to be set in this cleaning, namely the fraction of accidental images, which is set to 0.5%: this means that when this method is applied to pure NSB events, only 0.5% of the images are considered as shower images.

For images at the edge of the FOV, EventDisplay applies a likelihood fit assuming a Gaussian shape for the image in order to get a proper estimation of the image parameters [10] this method is important for high-energy showers that trigger further telescopes with respect to low energy ones. Another case in which this method could be applied is the divergent pointing in which more images are falling at edges of the cameras with respect to the normal pointing and instead of rejecting edge images due to their high leakage, they could be recovered and used to improve the reconstruction.

Direction estimation

The event reconstruction methods used for the creation of the IRFs for CTA are different between the two software packages but their common goal is to reconstruct parameters such as the impact point of the shower on the ground and the incoming direction in the sky.

The reconstruction of the direction in MARS uses the parameters from the Hillas parametrization of the camera images to get the best estimate of the direction reconstruction: this method minimizes the distance of closest approach between the reconstructed position and the directions given by the major axes of the ellipses in the cameras applying

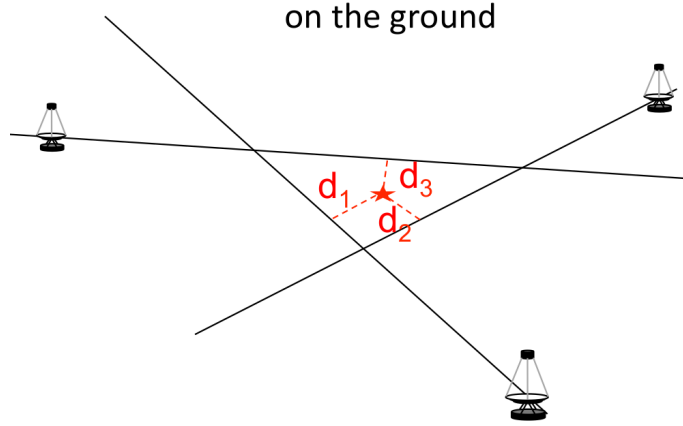


Figure 1.17: sketch showing the impact point reconstruction in MARS. The plane passing through each telescope is intersected with a horizontal plane corresponding to the ground and the reconstruction is done minimizing the distance between these lines from the reconstructed point.

a weight to each image. The weights are read from a Look-Up Table (LUT) which is filled using the inverse of the “distance of closest approach” (DCA) squared as a function of the size and the width/length parameters from the Hillas moments: the creation of those weights is done using roughly 20% of the simulated gamma-rays.

Some selection cuts are applied to the camera images used to build the LUT and at the analysis levels: images should have at least 50 photoelectrons and the ratio between the minor axis and the major axis should be between 0.1 and 0.6 (the eccentricity ranging from 0.995 to 0.8). In addition, at the analysis level, those images whose center of gravity (which is the average position of the ellipse) falls more than 80% of the camera radius from the center, are discarded due to their high leakage. The reconstruction of the impact point uses the intersection between the plane passing through each telescope and the $z = 0$ plane (the ground level) to find a line per each telescope in this common reference plane: the impact point is found as the point which minimizes the sum of its squared distances from these lines (see figure 1.17).

The height of the maximum emission of the shower, the h_{\max} parameter, is directly related to the energy of the primary particle and it can be used as a powerful gamma/hadron discriminator. The calculation of this parameter is done geometrically under the assumption that the shower maximum is projected in the center of gravity (CoG) of the camera ellipse. In the MARS analysis, a 3D line is drawn from each telescope using the position of the CoG of the ellipse and the tilt angle to get the direction of this line. The 3D lines from a stereoscopic observation are intersected with an horizontal plane at height h , generating a point in the plane per each line. h_{\max} is the height at which the spread between these point is minimum. The reconstruction with EventDisplay is done for each pair of telescopes, computing the distance D_{ij} between the telescope and the distance c_{ij} between the two image centroids: the ratio D_{ij}/c_{ij} gives h_{\max}^{ij} and the final value is a the mean of these values, computed using their image sized as weights.

EventDisplay performs the direction reconstruction intersecting the camera images

pair-wise and calculating for each pair a weight which is a combination of the angle between the major axis of the ellipses, their width/length ratio and their sizes: the final image is a simple weighed average over those values.

An improved method used in EventDisplay, which is also implemented in MARS but was not used for the CTA analyses, is the so-called DISP method, first reported in [12] and later on adapted and used in MAGIC [13], H.E.S.S. [14] and VERITAS [15]. This method uses either a LUT or a Boosted Decision Tree (BDT) to find a relation between the ellipse parameters (width, length, width/length, size, time gradient, asymmetry) and the true position of the source in the camera. Once the LUT (or BDT) is created, one can use the camera images to find the expected position of the source on the camera for each image. Each predicted position is combined to get an estimates of real shower direction. The reconstruction of the shower core is done in a similar way.

The disp method gives only an estimation of the distance with respect to the center of gravity along the major axis, but it's a really powerful method which works significantly better than the classical method especially for events with mostly parallel images, such as for large zenith angle observations or for large off-axis events [10].

Energy estimation

A Random Forest (RF) algorithm is used in MARS for the estimation of the energy of the primary particle [16]. The input parameters for this RF are both image parameters (size, width, length, etc.) and stereo parameters that have just been reconstructed, namely the impact parameter and the height of the maximum of the shower. Since in CTA each event is triggering a different number of telescopes of different type, it's not possible to train the RF on a event-by-event base, but a different RF is trained per telescope type. The energy and an associated error is estimated per each telescope and a weighed average, using the weights obtained from the RF, is done in order to get an estimation of the energy of the event and its associated error. An independent subset of the simulated gamma-rays is used for the training of the RF and it's then discarded since applying the RF to the same dataset used for the training will give biased results.

A simple energy estimator used in EventDisplay is a LUT: the median values $size_{ij}^{LUT}$ and 90%-widths σ_{ij}^{LUT} of the logarithm of the size parameter are used to fill the look-up table as a function of the distance d_i of the impact point from the telescope and the energy E_j of the primary particle.⁵ Then the energy E_k per each telescope are estimated using the logarithm of image size $size_k$ and the distance d_k from the telescope to the impact point on the ground. The final estimation is an average of the estimates from all the telescopes N_{tels} having a valid image:

$$E_{tot} = \frac{1}{N_{tels}} \sum_{k=1}^{N_{tels}} E_k \quad \text{with} \quad E_k = E_k(\log(size_k), d_k),$$

⁵The medians and the 90%-widths are used instead of the average and the RMS to reduce the impact of outliers in the distribution [17].

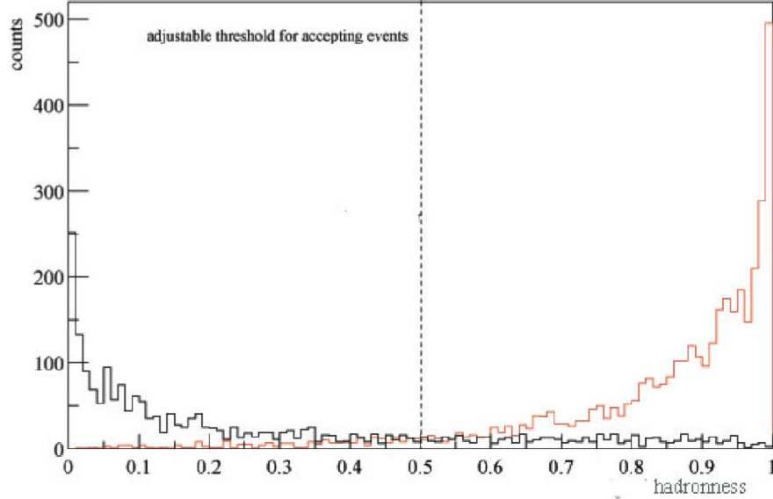


Figure 1.18: example of a hadronness distribution. The black curve on the left shows the hadronness for the analysis applied on the gamma rays, whereas the red line on right shows the hadronness for the protons. This plot includes events covering a large energy range. Plot from [18].

with $size_k$ and d_k being the size and distance to the impact point for the k -th telescope that are looked for in the LUT for the telescope type of the k -th telescope in order to find the corresponding E_k .

The BDT used for the direction reconstruction can also be used for the energy estimation, using the energy as regression target [10].

Gamma-hadron separation

In the MARS analysis, similarly to the energy estimation, a RF is grown using image parameters and stereo parameters together with the estimated energy from the previous step. The RFs are trained per telescope type using an independent set of gamma-rays and protons. The output of the RF per each telescope when applied to a dataset is a parameter ranging between 0 and 1, called hadronness, 0 being an event classified as a gamma-ray and 1 being an event classified as a proton (the total hadronness is a weighted average of the per-telescope value).

The hadronness is then computed on the real events, mixed gammas and background Monte Carlo events (protons and electrons) in the case of simulated data, and a hadronness histogram is produced. The hadronness cut is one of the parameters which is optimized in the next analysis steps, so that only those events below a certain threshold are kept: a small value guarantees that there's very few proton events mixed with the gamma-rays, while a higher values increases the number of protons wrongly accepted as gamma-rays, but also the statistics of the gamma-rays is much higher since very few gamma-rays are discarded (see figure 1.18).

In order to obtain a realistic event rate per each energy bin, the events from the Monte Carlo simulations must be weighted according to some real spectra. The protons

spectrum is parametrized as a power law from a fit of the ATIC data [19], the electron spectrum is obtained from a fit of the Fermi and H.E.S.S. measurements and the gamma-ray spectrum is assumed to be the one of the Crab Nebula as measured from the HEGRA experiment [20]. The comparison between the real spectra and those from the Monte Carlo simulations gives the weight to apply to the analysis done on the Monte Carlo simulations.

Concerning the EventDisplay analysis, similarly to what is done for the determination of the energy in the previous step, a LUT is created for both the length and the width parameter, for each telescope type, site, zenith angle, azimuth angle and NSB level, so that there will be a full set of LUT per each telescope type. A LUT for the length parameter (same applies to the width parameter) will be filled with the median values $\{\bar{l}^{LUT_n}\}_{ij}$ and the 90%-widths, $\{\sigma_{90}^{LUT_n}\}_{ij}$, of the distributions in the i -th size parameter bin and the j -th bin of the distance between the telescope and the impact point: each element of the n -th LUT (here written as LUT_n) will have two values then, namely the median and width of the distribution of all the size and distance pairs belonging to that element. Once the LUTs are done for each telescope type, zenith angle, azimuth angle, NSB level and for both the width and the length parameters, they are used in the estimation of the *mean reduced scaled width* (MRSW) and the *mean reduced scaled length* (MRSL) for all the events, using all the telescopes with valid images N_{tels} in each event:

$$MRSL = \frac{1}{N_{\text{tels}}} \left[\sum_{k=1}^{N_{\text{tels}}} \frac{l_k - \bar{l}^{LUT_n}(\Theta_k)}{\sigma_{90}^{LUT_n}(\Theta_k)} \right],$$

where $\Theta_k = (\text{size}_k, \text{distance}_k)$ are the input values for the k -th telescope and LUT_n is the LUT specific for that telescope type, azimuth angle, zenith angle, NSB level and site relative to that k -th telescope. Those parameters are useful to perform the gamma/hadron separation because while the parameters for the gamma-rays are centered around 0, hadronic showers produce images that are longer and wider: particularly, the MRSW parameter increases with increasing shower energy, making this a good parameter for the gamma/hadron separation [21].

After the *MRSL* and *MRSW* have been computed, a BDT (using the Toolkit for Multivariate Data Analysis (TMVA) implementation from the ROOT framework) is trained in different energy bins and in different off-axis bins. The training of those BDTs is done using a mix of gamma-rays and protons where every event's type is known and the parameters used for the training of the BDTs are: MRSW, MRSL, emission height, the size of the second-most bright image, the telescope-impact point distance and the χ^2 -values for the emission height and the energy (see figure 1.19). Before the BDTs are trained, some selection cuts are also applied to the training parameters.

Cuts optimization

After the steps above, the MARS analysis continues with the optimization looking for the combination of parameters giving the best possible point-source gamma-ray flux sensitivity. The most important parameters are: reconstructed direction (expresses with square

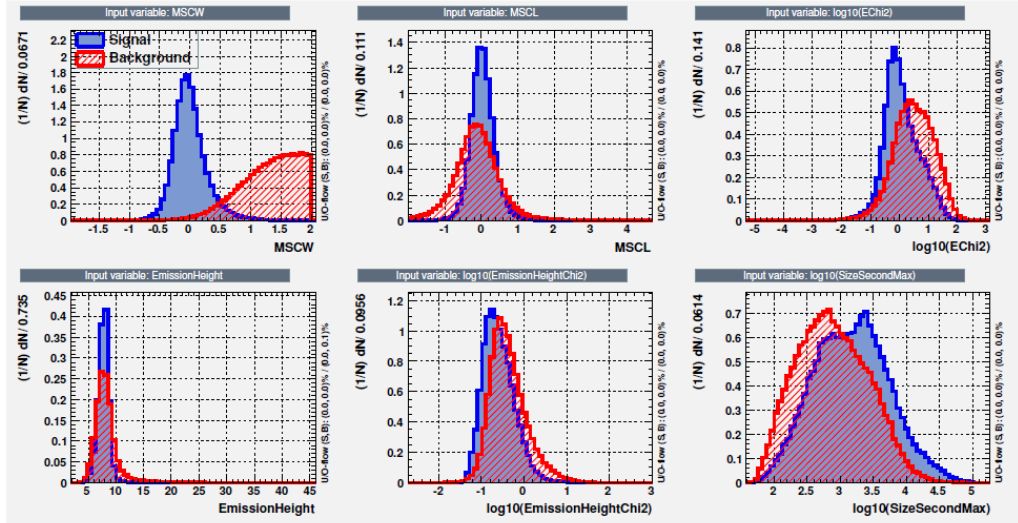


Figure 1.19: histograms for the various parameters (Mean SCAled Width, Mean SCAled Length, χ^2 -values for the energy, emission height, χ^2 -values for the emission height, size of the second-most bright image) used for the training of the BDTs to be later used for event classification in EventDisplay. While the blue histograms are showing the distribution of the “Signal” events, the red ones show the distribution for the background events. The larger is the difference between the blue and red distributions, the most that parameter is useful for the gamma/hadron discrimination. The events have a reconstructed energy of roughly 10 TeV (figure from [10]).

the angular distance between the reconstructed position and the source position, θ^2), reconstructed energy, hadronness and telescope multiplicity.

In each of the five bins per decade in reconstructed energy, cuts in the hadronness and in the θ^2 are used, ranging from 0.5 to 0.95 for both parameters, and also a cut in the multiplicity is applied with at least 2 to 9 telescopes triggered in an event. Each set of parameters in this multidimensional space is used to compute the sensitivity with the optimization criterium of best possible gamma-ray flux sensitivity to a point-source, the conditions being: at least 10 gamma-rays, significance greater than 5 according to [22] and a number of gamma-rays greater than 5% of the residual background.

EventDisplay has the same optimization goal, using the same spectra presented in the previous section to obtain the event rate from the Monte Carlo simulations, but the cuts are applied to the BDT output. This output ranged from -0.5 to 0.5 and the optimal BDT cut is selected at the value giving the best significance for point-like source with a Crab-like shaped SED.

The optimal BDTs cuts, in this case, are also optimized according to the observation time and are determined for each energy bin separately.

1.3.5 IRF usage

It’s really important to keep in mind how those IRFs are generated when using them for the particular physics case: since those IRFs are created with the goal of having the best sensitivity, while fulfilling the minimum requirement for angular and energy resolution, they where not optimized for other studies.

If those are used to compute the sensitivity of very faint sources, very bright ones, extended sources or even for studies requiring a really good angular resolution, as in the case of line searches for the Dark Matter studies, the resulting performances of CTA towards that specific science topic will be either underestimated or overestimated.

1.4 CTA Computing

As already said before, the selection of the sites and the optimization of the telescopes layout has been possible thanks to massive Monte Carlo simulations starting with the interaction of the primary particles with the atmosphere to end with the creation of signal in the camera. The Monte Carlo productions have been created using computing centres, operated as a distributed computing infrastructure, spread across several states belonging to the European Grid Infrastructure, the EGI, with 20 sites spread over 7 countries (full list available at [7]).

In order to manage CTA production activities over a distributed computing infrastructure, where computing centres share the global storage and the computing loads, a prototype based on the DIRAC framework, called “CTA-DIRAC”, has been deployed on the EGI. The framework is used to perform the Monte Carlo simulations and to analyse the results so that both raw datasets and higher level datasets are available to the users: in order to connect to the computing grid, submit jobs and download data, a certificate must be requested.

The GRID has been used in this thesis for the creation of the Monte Carlo simulations for the estimation of the performances of CTA in the divergent pointing observation mode. The creation of these simulations on a local computing farm would have required some TB of storage and several days of computing time. The simulations were done in just a day on GRID with basically no impact on the global storage, due to the small size of this production.

Chapter 2

Science with CTA

CTA is going to be the largest array of Imaging Atmospheric Cherenkov Telescopes with a sensitivity from five times up to 20 times higher with respect to existing similar experiments, thanks to the large number of telescopes (more than 100) built across two sites, one in the northern hemisphere, in La Palma (Canary Islands, Spain), and one in the southern hemisphere, in Cerro Paranal in Chile. The prototype for the Large Size Telescope is in the commissioning phase at Roque de Los Muchachos (La Palma), whereas many other prototypes of the other telescopes types are being built and tested by the CTA Consortium which includes more than 1500 scientists and engineers from 200 institutes distributed in 31 countries.

In this chapter I'll review the scientific goals of CTA and their importance as drivers for the design of the telescopes themselves. In the first section I'll give an overview of the science goals and in the following sections I'll go a bit more into the details of two phenomena that are part of Key Science Project for transient sources. I'll describe in depth the physics of Gamma Ray Bursts and Gravitational Waves showing the state of the art of these phenomena and which are the prospects for the observation of such phenomena with CTA: I'll go into the details of these phenomena because I've been working on the two related CTA Consortium papers.

2.1 Science themes

2.1.1 Key Science Projects

The CTA Observatory will be operated as an open observatory, accepting proposals from the external community and releasing all data as public after a proprietary period of one year. The Consortium has prepared a series of Key Science Projects (KSPs) that are going to use 40% of the available observing time in the first 10 years of observations. The KSPs covers the Galactic and the Extragalactic domain, with deep surveys of the Large Magellanic Clouds and of the Galactic Centre, observation and monitoring of transient sources, search for PeVatrons, star-forming regions on a wide range of scales and the

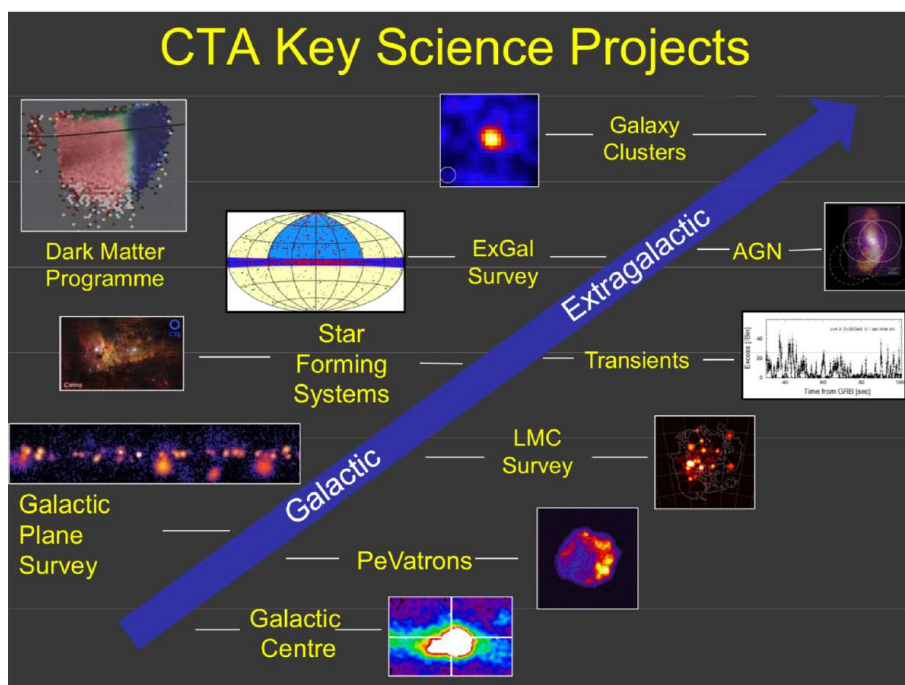


Figure 2.1: the Key Science Projects of CTA listed according to their distance from us. Image from [23].

Perseus Galaxy Cluster: in addition to this, as mentioned in the previous section, there is a Dark Matter programme.

I will not enter into the details of each KSP (fully explained in chapters 4 to 13 of [24]), but I will present two topics that I've been working on during my thesis, namely the Gamma Ray Bursts and the Gravitational Waves, that are part of the Transient KSP together with Galactic transients, transients observed in other wavelengths and high-energy neutrino transients, based on alerts from neutrino observatories.

There is a wide range of major questions that CTA is going to address [24] which can be grouped in three main themes that I'm briefly going to present here below.

Origin and Role of Relativistic Cosmic Particles

The discovery of the extraterrestrial origin of Cosmic Rays (CRs) dates back to the beginning of the 20th century and from that time until now many experiments have observed these energetic particles from the lowest energies to the highest energies (see spectrum of CRs in figure 2.2). Many questions have been raised concerning the sites and the mechanisms responsible for the acceleration of these particles to the highest energies and which role CRs play in star formation and galaxy evolution, and in order to address these questions, CTA will observe astrophysical systems with the highest resolution ever achieved in the GeV–TeV domain. In 2013 the satellites *Fermi* and *AGILE* have confirmed Supernova Remnants as sources of high-energy protons [27], but it's still unclear whether Galactic CRs are predominantly produced by young SNRs. The gamma-ray spectra of the middle-

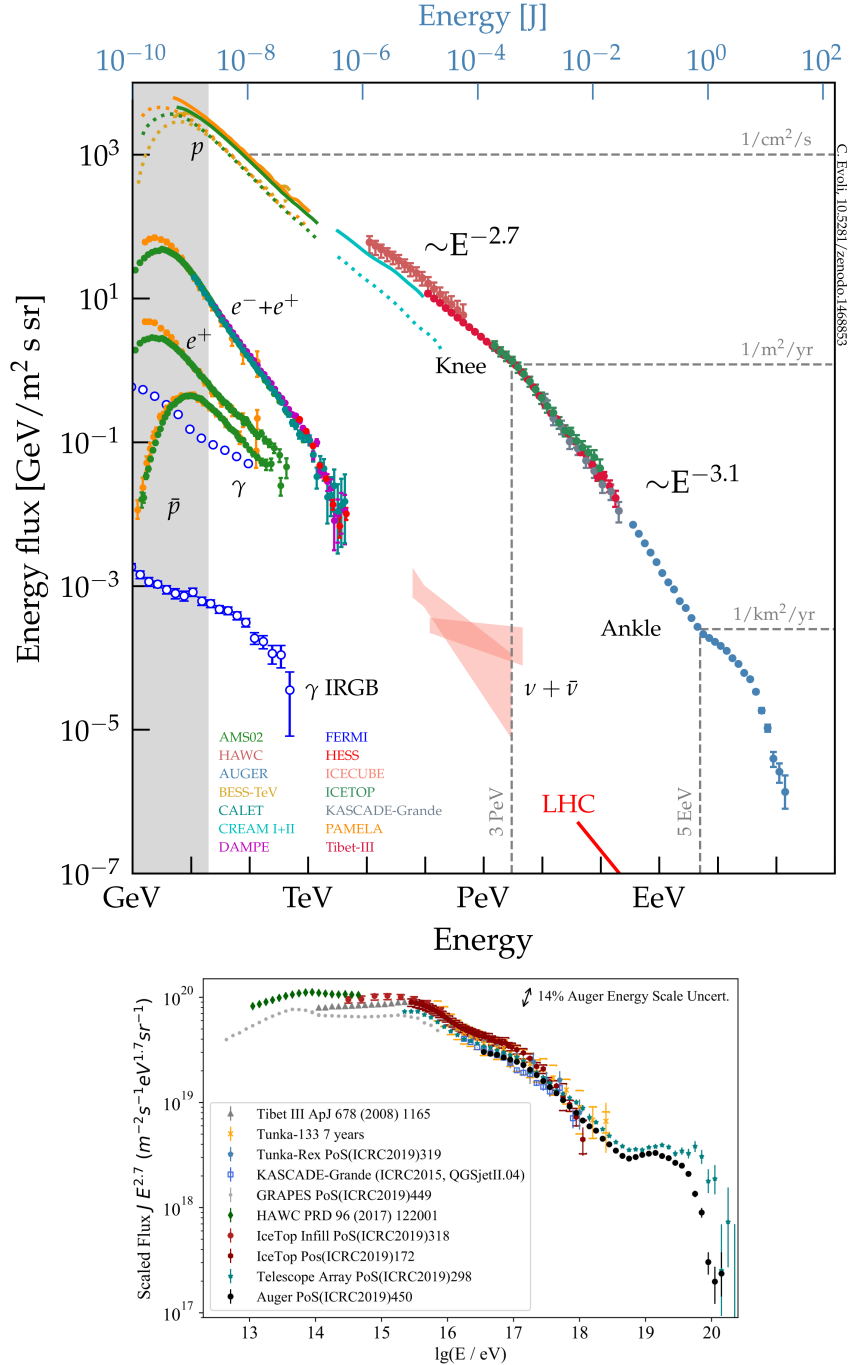


Figure 2.2: *Upper panel*: cosmic-ray energy spectrum measured by several experiments (figure from [25]). *Lower panel*: highest energy range zoom for the spectral energy distribution of high-energy cosmic rays (multiplied by $E^{-2.7}$) obtained from air-shower measurements (figure from [26]). The energy range covered by H.E.S.S., reported in red in the upper panel, is similar to the one that will be covered with CTA.

aged SNRs IC443 and W44 observed by AGILE and Fermi have shown a cut-off at roughly 100 MeV reminiscent of the π^0 bump that provides a direct proof for the hadronic origin of the gamma rays. The maximum CR energies inferred from these spectra is much lower than the “knee” and their gamma-ray spectra at VHE is very low, suggesting that only a younger SNR population could be a potential PeVatron candidate, a source capable of

accelerating particles at the PeV scale. Therefore the search for sources emitting particles at the PeV energies is a key item in the observation of Galactic sources with CTA.

Thanks to the Galactic and Extragalactic surveys, with deeper observations of nearby galaxies and clusters and to precise measurements of bright nearby sources with high spatial, temporal and spectral resolution, CTA will deepen our knowledge on cosmic rays acceleration sites and the mechanisms at play. Known TeV-emitting sources, such as Pulsar Wind Nebulae, gamma-ray binaries, colliding-wind binaries, massive stellar clusters, starburst galaxies and active galaxies will be observed in detail.

A map of the extended emission around many gamma-ray sources, such as the Supernova Remnant RX J1713.7-3946, in search for an energy dependent morphology associated either with protons (photons from the decay of π^0 s generated by accelerated protons colliding with the surrounding gas) or electrons (low-energy photons upscattered by high-energy electrons), will allow CTA to separate the emission into an hadronic and a leptonic component [24] (see prospects for RX J1713.7-3946 in [28]).

The observation of gamma rays from sources of our Galaxy and from other nearby sources, such as the Large Magellanic Cloud or the Andromeda Galaxy M31, is an important target because it's possible to understand where CRs interact in the interstellar medium (ISM) observing the distribution of gamma rays produced from their interaction. CRs are believed to be an important regulator of the star-formation process, through the ionisation of the ISM material which affects the molecular cloud structure and star formation. Supernova explosions are important for the enrichment of the ISM with heavy elements, and since SNRs are also responsible for producing CRs that interact with the ISM, these sources can suppress or enhance the star-formation [24].

It's necessary to study the VHE gamma-ray emission from nearby sources to understand how the acceleration and propagation of CRs works. These gamma rays are left unaltered between the production site and the Earth, which is not the case for high redshift sources due to the interaction with the Extragalactic Background Light (EBL, see appendix A), which has the effect of absorbing the flux at the highest energies for sources at high redshift. Simulations of galaxy formation coupling CRs to the magnetohydrodynamics have shown how CRs can launch powerful galactic winds, reducing the available amount of gas for star formation and, on scales of the galaxy clusters, the effect of the CRs is to mitigate the star formation of the brightest clusters [29].

Probing Extreme Environments

Particle acceleration is usually linked to explosions, relativistic outflows and extreme environments, such as those close to black holes and neutron stars, where dense objects produce a huge gravitational pull on the surrounding material.

Black holes, especially the supermassive black holes (SMBHs) at the center of active galactic nuclei (AGN), are characterised by the accretion of material and the subsequent emission of collimated relativistic jets. The particle acceleration happens both near the

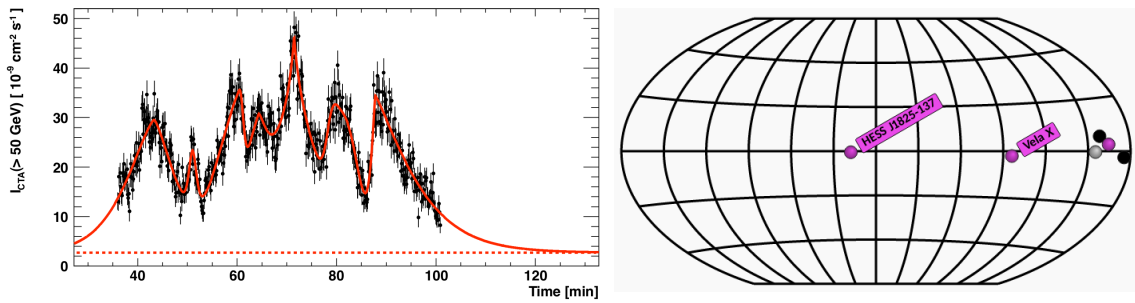


Figure 2.3: *Left panel:* figure 1.7 from [24] representing the simulated light curve for the 2006 flare of PKS 2155-304, which shows the fast variability of AGNs. *Right panel:* Position in a galactic reference frame of the two sources that CTA will observe in order to probe the environment surrounding them. Other 4 sources, labelled as TeV-halo are reported in this map on the right: the magenta one is Geminga and the other three have been discovered by HAWK. Figure generated from the TeVCat 2.0 website (<http://tevcat2.uchicago.edu/>).

centre of the black hole, which explains the variability over very short temporal scales (see left panel on figure 2.3), and along the AGN jets, with AGNs being the most likely acceleration sites for the UHECRs, which are cosmic rays with energies reaching up to 10^{20} eV. Since there’s no strong evidence proving this picture, the AGN observation program of CTA is going to be really important, together with observations in other wavelengths, to understand the physical processes at work close to black holes. These programs will allow to disentangle leptonic and hadronic emission scenarios and to study the jets power and dynamics and measure the magnetic fields in those extreme environments: a powerful probe of the weak intergalactic magnetic fields (IGMF) is the observation of secondary components, produced by the primary TeV, as a delayed emission relative to the primary gamma ray (the “pair echoes”) or an extended emission around the primary source (“pair halos”) [30]. The SMBH Sgr A* at the center of our Galaxy and the system Cygnus X-1 will be observed with unprecedented sensitivity from CTA as part of the Galactic Plane Survey and the Galactic Center KSPs: this is especially motivated from recent observations with H.E.S.S. of the Galactic Centre [31].

The Galactic Plane Survey will also search for emission of VHE gamma rays from pulsars, such as the young and energetic HESS J1825-137 and the Vela pulsar (see right panel of figure 2.3) and the Crab Nebula, together with other pulsars seen at high energies or previously unidentified sources that CTA might reveal while looking in depth at the Galactic plane.

Furthermore, VHE photons can escape from systems in which UV or X-rays are absorbed and they can be used to probe these extreme environments, especially their spatial and time scales and energetics.

Exploring Frontiers in Physics

The energy range that CTA will be exploring allows the investigation of many topics of fundamental physics. The nature of dark matter is one of the major open questions in

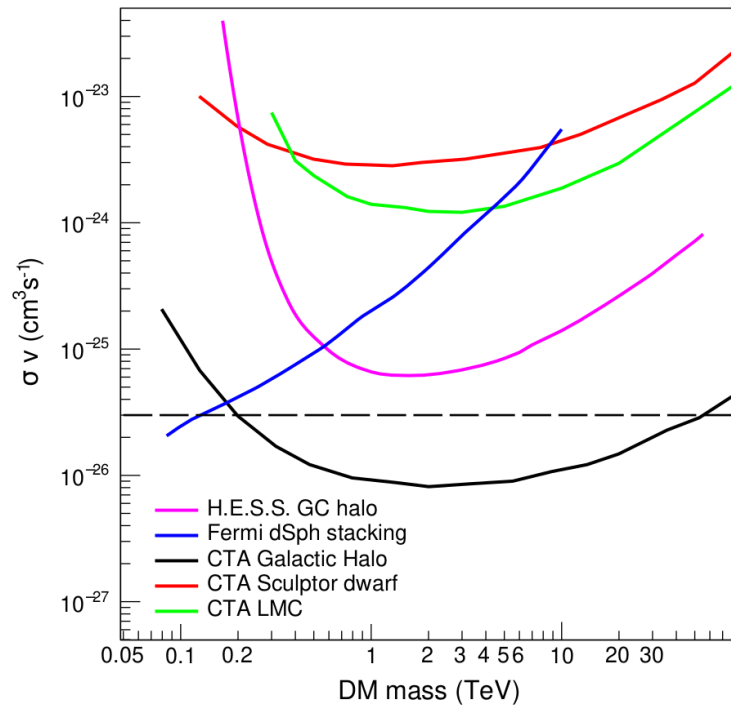


Figure 2.4: CTA sensitivity to a WIMP annihilation signature as a function of WIMP mass. The dashed horizontal line indicates the likely cross-section for a WIMP which is a thermal relic of the Big Bang. Figure 1.8 from [24].

modern physics, with its presence and density derived from measurements of the power spectrum of the Cosmic Microwave Background radiation (the Planck mission [32] reported a measurement of the dark matter density parameter Ω_c equal to 0.2589 ± 0.0057): while the role of dark matter as seed for the formation of cosmic structures is clear, it's not yet clear what is dark matter made of.

In the most popular scenario for cold dark matter (the actual cosmological model is the Λ CDM model, Lambda Cold Dark Matter), the particles making up this matter are massive particles with a small cross section, called Weakly Interacting Massive Particles (WIMPs), which comprise a large class of non-baryonic candidates with masses between few tens of GeV to few TeV and with an annihilation cross-section of the order of the weak interaction. Those numbers give a self-annihilating cross-section that would imply a relic abundance for dark matter, the present quantity of dark matter remaining after the Big Bang, to be close to the currently observed value.

With the dedicated observation of the region surrounding the Galactic Centre, CTA is expected to reach a velocity-averaged annihilation cross-section of $\sim 3 \times 10^{-26} \text{ cm}^3 \text{ s}^{-1}$ for a dark matter mass in the range 200 GeV – 20 TeV, which is as limit not possible to achieve with any other current instrument (see figure 2.4). Other targets that will be searched for dark matter are the satellite galaxies of our Milky Way, such as the Large Magellanic Cloud and the dwarf galaxies, such as Sculptor, expected to contain much more dark matter than large spirals.

The improved energy resolution of CTA with respect to existing instruments will allow CTA to detect possible emission lines in the spectra coming from the annihilation of two dark matter particles into two gamma rays. If no emission compatible with a dark matter annihilation signal will be measured with CTA, the WIMP paradigm for a Λ CDM scenario will be more constrained.

Together with dark matter searches, CTA will search for new physics beyond the Standard Model looking for axion-like particles (ALPs), evidence of Lorentz invariance violation (LIV) and in quantum gravity effects on space-time at the Planck scale. Axions are expected to convert into photons (and viceversa) and not interact with the EBL while in the axion state. The observation of a high energy photon from a source at high redshift, such as a blazar or a GRB whose flux is expected to be low at high energies due to the interaction with the EBL, might be a hint of ALPs. Quantum gravity effects, through a modification of the energy-momentum relation with a corrective term, may induce time delays between photons at different energy, which might be inspected from high statistics measurements of the spectra of GRBs and blazars (see limits from *Fermi* in [33]).

2.2 Gamma-Ray Bursts

2.2.1 Introduction

Gamma-Ray bursts (GRBs) are the most luminous explosions in the Universe, liberating as much as $10^{52} - 10^{54}$ erg of isotropic-equivalent energy during a brief period of 0.01–1000 s, primarily in the MeV band. They also manifest rapid and irregular variability on timescales down to sub-millisecond levels [34].

These sources have been discovered in the late 1960s by the U.S. Vela nuclear test detection satellites, a constellation of 12 satellites equipped with X-ray, neutron and gamma-ray detectors.

On July 2, 1967, at 14:19 UTC, the Vela 3 and Vela 4 satellites detected a flash of gamma radiation which neither had the expected signature of the nuclear explosion or could be explained by a solar flare or a supernova: this was the first detection of this kind of sources, characterized by an intense emission and a rapid variation.

A big improvement came from the launch of the Compton Gamma-Ray Observatory (CGRO), in 1991, which had four instruments onboard (BATSE, OSSE, COMPTEL and EGRET), sensitive from 20 keV to 30 GeV. The Burst and Transient Source Experiment (BATSE), dedicated to the detection of GRBs, observing the sky from 20 keV to 8 MeV, detected more or less one GRB per day over the 9-year CGRO mission. The distribution of the T_{90} , the interval going from T_5 to T_{95} that are the times at which the 5% and the 95%

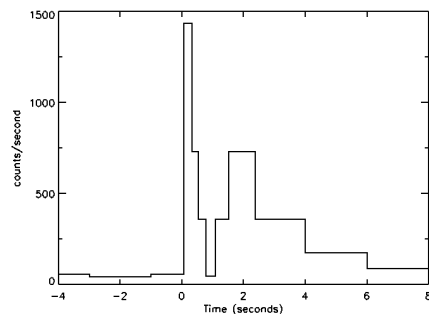


Figure 2.5: First gamma-ray burst detected from the Vela satellites. (Credit: R. Klebesadel, I. Strong & R. Olson).

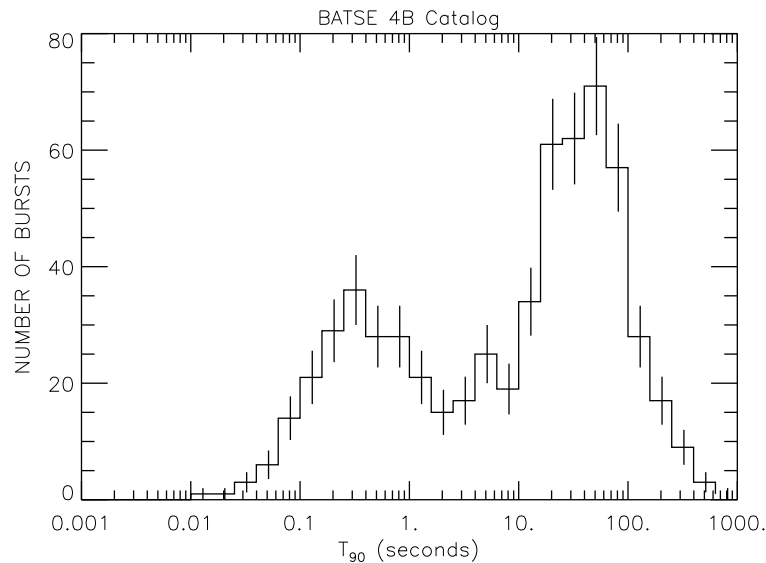


Figure 2.6: Distribution of T_{90} for GRBs of the fourth BATSE catalog (figure from [36]). The data used for the calculation are the BATSE 4 energy channel discriminator data. Lightcurves used for the calculation of T_{90} are integrated over all 4 channels ($E > 20$ keV).

of the counts are detected, for 2000 GRBs observed by BATSE is reported in figure 2.6 and shows a double-peaked distribution. This observation led to the division of GRBs into two types [35]: the short GRBs, typically lasting less than 2 seconds, and the long ones, lasting more than two seconds (see figure 2.6), with the spectrum of short GRBs being harder with respect to the one of long GRBs.

Furthermore, the analysis of the GRBs observed by BATSE [37] showed that these sources are isotropically distributed in the sky and that, while observing the histogram of their cumulative distribution, the number of faint sources deviates from what one would find if these sources were considered uniformly distributed in an Euclidean space (the total number of sources an observer sees with flux above f should scale as $f^{-3/2}$). These evidences convinced scientists that the characteristic distances of these objects were much larger than the size of the local group of galaxies [38].

The confirmation of the cosmological origin of GRBs came from the Italian-Dutch satellite for X-ray astronomy called Beppo-SAX: the satellite, operating between 1996 and 2002, had a wide spectral coverage, from 0.1 keV to 300 keV, a relatively large area and a good (relative to that time) resolution in energy. This discovery was possible thanks to the detection of the rapidly fading emission belonging to the *afterglow* of GRB970228,¹ in X-ray band (by Beppo-SAX) and in the optical band (by optical telescopes); the detection of this emission was done 8 hours after the burst occurred and deeper observations with the Keck telescopes led to the determination of the redshift of the source to be 0.695 [39], giving this GRB an isotropic emission of 5.2×10^{51} erg.

¹The name of a GRB is constructed from the day in which it was discovered: in this case, GRB970228 was seen in the 28th of February, 1997. A capital letter is attached at the end of the name, like GRB080916C, so that the first GRB observed in a day is given the “A”, “B” for the second GRB, and so on.

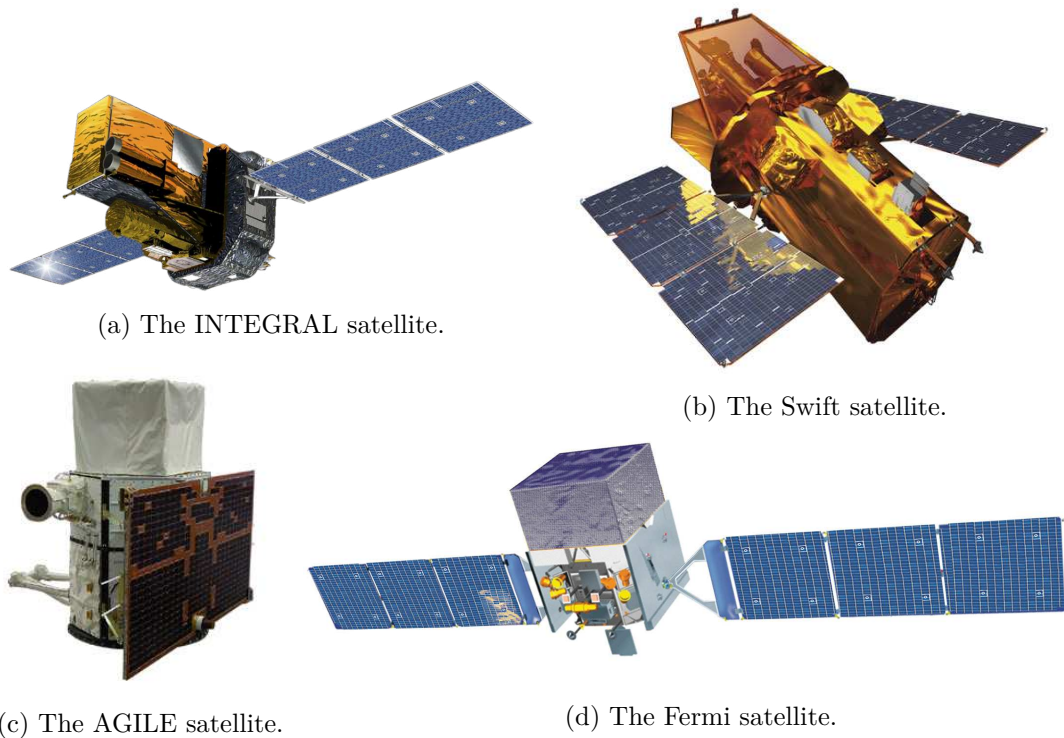


Figure 2.7: The four major space-based observatories operative nowadays from the UV band up to the high energy regime.

Nowadays there are four major space-based observatories searching for GRBs from the UV band to the high energy regime (see figures 2.7): the older one is the INTEGRAL (INTERNational Gamma-Ray Astrophysics Laboratory) satellite (figure 2.7a), launched in 2002, capable of detecting the GRBs in the low-energy gamma-ray and X-ray using the INTEGRAL Burst Alert System (IBAS). This system is based on data from the ISGRI detector of the IBIS imaging instrument and from the SPI anticoincidence system of INTEGRAL.

In 2004 the SWIFT satellite was launched, equipped with three instruments onboard, that are capable of observing GRBs in the gamma-ray, X-ray, ultraviolet and optical wavelengths. Thanks to its multiwavelength instruments and its fast slewing, Swift has been able to observe more than 1000 GRBs from the beginning of the mission.

The satellite AGILE (Astro-Rivelatore Gamma a Immagini Leggero) was launched by ASI in 2007, reporting 85 GRBs observed with the Mini-Calorimeter (350 keV to 100 MeV) in 18 months of observations [40] (1 GRB/week), with 10 GRBs emitting high-energy photons.

The *Fermi* satellite is the most recent satellite, launched in 2008 by the NASA, and has two instruments: one is the Gamma-ray Burst Monitor (GBM), which consists of 14 scintillation detectors (12 NaI crystals for the 8 keV–1 MeV range and two BGO crystals, sensitive from 150 keV up to 30 MeV) and it is used for the detection of GRBs across the whole sky not occulted by the Earth. The second instrument, the Large Area Telescope

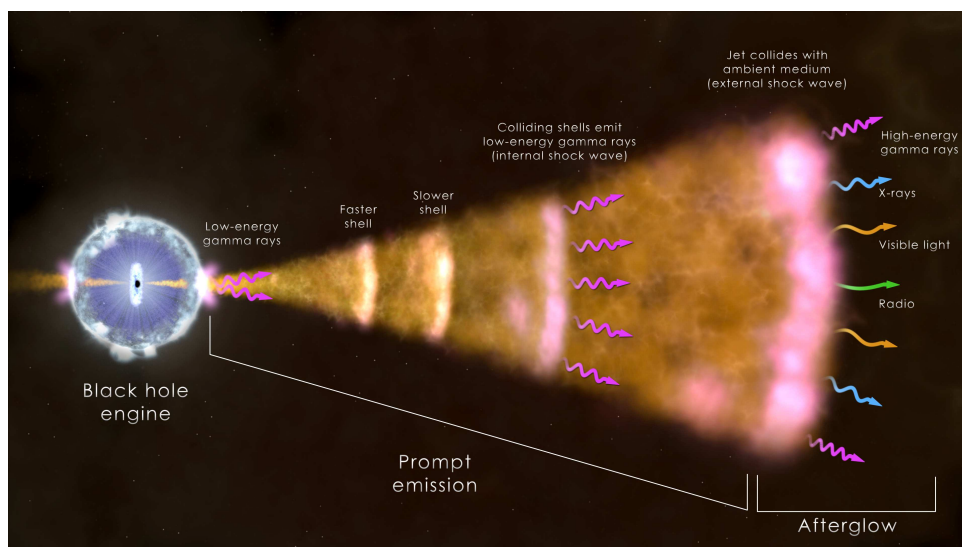


Figure 2.8: prompt and afterglow phases of GRBs (from <http://fermi.gsfc.nasa.gov/science/etev/grbs/>).

(LAT), is sensitive to higher energies than GBM, from 20 MeV to 300 GeV; this instrument uses the conversion of an incoming photon in an electron-positron pair which are tracked across the silicon layers down to a calorimeter which measures the energy of the pair. GBM has detected 1405 GRBs from 2008 up to the middle of July 2014, as reported in the third GRB catalog of GBM [41], whereas LAT has detected 186 GRBs from 2008 up to the 4th of August 2018 [42], with 91 showing emission in the 30–100 MeV energy range and 169 detected above 100 MeV.

2.2.2 Phases of a GRB

There are two main phases in the development of a GRB: the first phase, called **prompt** phase, lasts typically only from a few seconds up to tens of seconds and the second phase, called **afterglow**, lasts longer than the prompt and in some cases the optical follow-up was possible for many months.

Prompt phase

The light curves for the prompt phase are very different from one burst to another and there isn't an analytical model that is able to describe them. The evolution of the parameters describing the spectrum of a GRB is also an important factor to understand the physics of GRBs.

GRB spectra are traditionally fitted with a mathematical function, which is known as

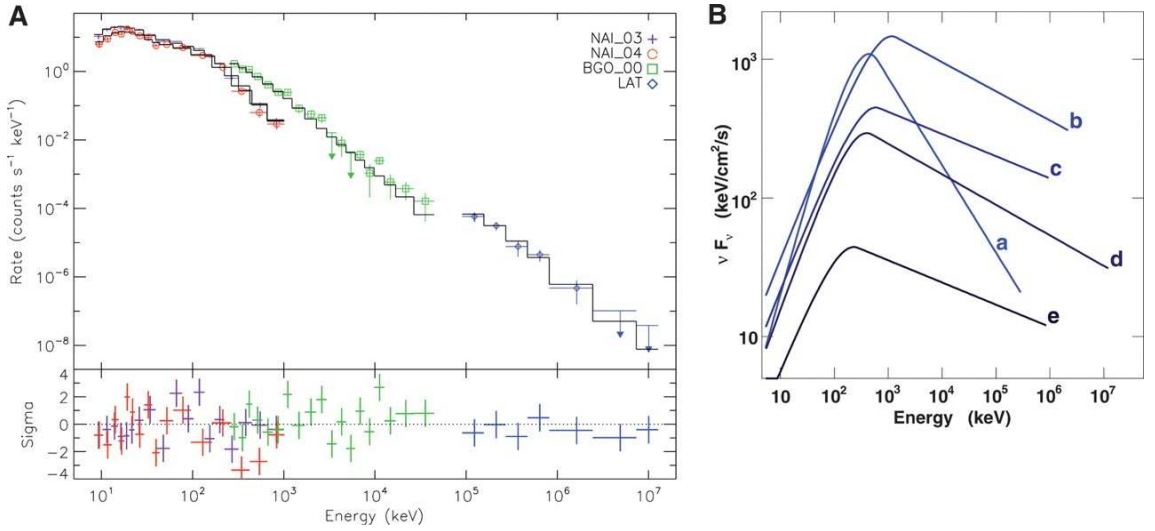


Figure 2.9: Spectra of GRB 080916C at five subsequent time intervals (*a* starts at the trigger and end after 3.6 s, whereas *e* starts at 54 s and end at 100 s after the trigger) are fitted with a “Band” function. (a) Count spectrum for NaI, BGO, and LAT in time bin *b*. (b) The “Band” model spectra in νF_ν units for all five time intervals showing the evolution of the spectrum over time. Figure from Abdo et al. [44].

the “Band” function, named after David Band who proposed this model [43]:

$$N_{\text{ph}}(E) = A \begin{cases} \left(\frac{E}{100 \text{ keV}}\right)^\alpha \exp\left(-\frac{E}{E_0}\right) & E < (\alpha - \beta)E_0 \\ \left[\frac{(\alpha - \beta)E_0}{100 \text{ keV}}\right]^{\alpha - \beta} \exp(\beta - \alpha) \left(\frac{E}{100 \text{ keV}}\right)^\beta & E \geq (\alpha - \beta)E_0 \end{cases} \quad (2.1)$$

where $N(E)dE$ is the number of photons in the energy bin dE , α is the spectral slope at lower energies, β is the high energy spectral slope, E_0 is the break energy and A is an overall normalization. This function, used to fit the spectra in the keV – MeV region, often referred to as “Band model”, has become a standard in this field. However, this fitting function doesn’t come from any physical mechanism but is only a mathematical function which is able to fit well the spectra of the GRBs, with only 4 parameters (see example in figure 2.9).

The spectra are usually plotted as SED (Spectral Energy Distributions) which corresponds to $E^2N(E)$ or νF_ν , which exhibit a peak at the energy $E_p = (2 + \alpha)E_0$. Before the Fermi satellite was launched, it was believed that the prompt spectra of the GRB was described only by the Band function, but later on it was realised that together with the non-thermal Band function, additional components are needed in order to provide a better description of the spectra. There can be an additional power law component, with or without a cutoff and also a black body thermal component.

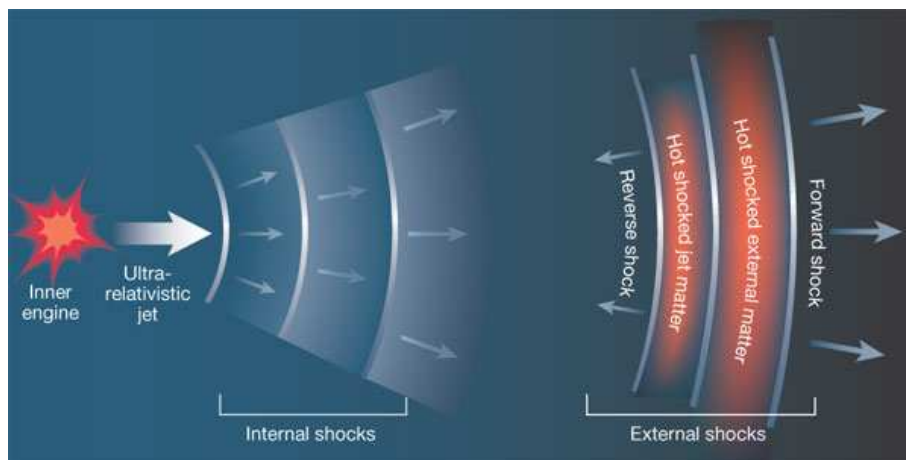


Figure 2.10: internal and external shock models for GRBs. The internal shocks produces a vast amount of gamma-rays and when the jet is slowed down by surrounding matter, external shocks are created with the forward propagating into space and the reverse shock that is reflected back against the relativistic flow, producing heating up the material, producing the afterglow of the GRB. From T.Piran, *Astronomy: Glowing embers*, *Nature* **422** (2003) 268.

Afterglow phase

The first afterglow emission from a GRB was observed in the optical and X-rays (by BeppoSAX) in GRB970228, which lead to the confirmation of the GRBs as sources placed at cosmological distances, confirming what had been theorized by Meszaros and Rees [45].

The first GRB to have an afterglow emission also in the optical and radio was GRB970508: it was observed in the X-rays by BeppoSAX and BATSE, while the afterglow emission of the burst was observed few days later. The radio observation started with the Very Large Array four hours after the discovery of the burst and were resumed five days after the discovery of the GRB and a radio signal was observed coming from the burst's position at different wavelengths, which confirmed the radio emission from the afterglow of a GRB. The 4-weeks-long observation of the radio afterglow emission lead to the first estimate of the source size and of the expansion rate, showing an expansion comparable to that of light [46].

2.2.3 GRB modelling: the fireball model

One of the widely discussed models for GRBs is the so called **hot fireball model**. This model was suggested in its currently used form by Paczyński and Goodman in 1986 (a first formulation appeared in [47]) when Paczyński realized that GRBs might have a cosmological distance so that the predicted luminosity is $\sim 10^{51}$ erg/s and it is produced within a small radius of $\sim 10^2$ km (from variabilities of the lightcurve of the order of a few milliseconds). From this spatial and energetic arguments, a temperature of $\sim 10^{10}$ K raises, so that electron-positron pairs coexist with photons in thermal equilibrium.

A hot fireball might be produced either by the merging of two compact objects or by the collapse of a massive star, and while at first the energy is mainly radiative, later on it

is converted into kinetic energy of protons, that are accelerated to a high Lorentz factor.

According to this model, the energy per proton is of order of 10^2 GeV with much of this energy being initially in photons, in relativistic electron-positron pairs and neutrinos. The radius where the fireball is produced is set by the size of the compact object formed in these explosions, believed to be either a black hole or a millisecond magnetar and as the fireball undergoes adiabatic expansion, the energy of photons and electron-positron pairs is transferred to protons which are accelerated to a high Lorentz factor. The kinetic energy of the outflow is converted back to thermal energy and radiated away as gamma-rays at some large distances from the place where the fireball is produced [38].

The black hole engine produced by the progenitors has been fed, for a short time, by the infall of the rotating gas which was left in the medium surrounding the progenitors and while some of this material is accreted, a larger fraction will be ejected in a jet along the axis of rotation. The rotating material will lead to very intense magnetic fields which extract the rotational energy of the black hole and pump it into the jet, which then becomes highly relativistic and collimated (5 to 10 degrees wide).

The initial large particle density decreases with increasing distance from the centre and at the photospheric radius, where the photon mean free path becomes larger than the dimension of the jet, the previously trapped photons are now free to escape.

One process that is able to dissipate energy into random energy of particles is the dissipation of energy beyond the photosphere in shocks. Two types of shocks are possible: either there are **internal shocks** within the jet itself or there are **external shocks**, due to the deceleration of the jet and its encounter with the external matter [48].

The Fermi mechanism accelerates the electrons to relativistic velocities thanks to their motion across the shocks, producing a population of electrons with a power law spectrum. The electrons will then produce a non-thermal photon spectrum either interacting with magnetic fields producing synchrotron radiation or interacting via Inverse Compton on low-energy photons.

When the deceleration begins, the external shock is accompanied by a reverse shock: this predicts a prompt optical emission, which has been detected with experiments such as the robotic telescope ROTSE (Robotic Optical Transient Search Experiment) which performs optical follow-up of GRBs.

As the deceleration of the jet continues due to the encounter of more and more external matter, the bulk Lorentz factor of the external shock decreases and a long lasting, fading X-ray, optical and radio afterglow is produced.

In a leptonic scenario electrons are accelerated in shocks, producing high-energy photons via synchrotron emission or Inverse Compton processes. It was shown by Kumar et al. in [49] that it is not easy to produce photons with energy larger than ~ 50 MeV (in the plasma comoving frame) via the synchrotron process. It is possible that the highest energy photons (with energies greater than 5 GeV, for a typical Γ of 100) detected by Fermi LAT from GRBs are produced via Inverse Compton from the interaction between high-energy electrons and lower-energy synchrotron photons [50].

Hadronic model

Protons are also accelerated in shocks and due to their smaller radiative loss rate they can reach much higher energies thus being able to contribute to the gamma-ray emission: the maximum synchrotron photon energy for protons (accelerated in shocks) is a factor $m_{\text{proton}}/m_{\text{electron}}$ larger than that for electrons [38]. This means that instead of a maximum energy of ~ 50 MeV (comoving frame) for electron-synchrotron photons, the proton synchrotron process can produce photons of energy 100 GeV (in the jet comoving frame). Therefore, when photons of energy larger than $\sim 10^2 \Gamma$ MeV are detected from a source, proton synchrotron process is thought to be a valuable mechanism to account for such energies. Due to the low radiative efficiency of the proton-synchrotron process, the energy required to produce a photon flux at the level observed by the Fermi satellite above few GeV is found to be in general too high [51].

The emission of gamma-rays from protons can also come from an indirect way: high energy protons can produce positrons and electrons with a large Lorentz factor via photo-pion and Bethe-Heitler processes, both involving collisions between energetic protons and photons to pions decaying into positrons and neutrinos (photo-pion) or produce e^\pm directly (Bethe-Heitler). These two processes, although inefficient for producing high energy electrons with respect to the Fermi mechanisms (electrons accelerated in the shockwave), they can become important at higher energies [38].

The observation of GRBs at the highest energies is very important in order to discriminate between a leptonic and an hadronic emission models [52].

2.2.4 HE emission from GRBs

Here I will focus mainly on the high energy radiation, above 100 MeV, without discussing the optical, X-ray and radio counterpart of the afterglow emission.

The First Fermi-LAT GRB Catalog (1FLGC) [53], with 35 GRBs detected, has brought some observational discoveries regarding the high-energy emission from GRB: (1) GRBs have a hard spectrum and many of them are not fitted with the Band function, requiring a power-law at higher energies for four GRBs, (2) the emission above 100 MeV is delayed up to 40 seconds, with an average of few seconds, with respect to the keV–MeV energy range and (3) the emission above 100 MeV lasts longer than the keV–MeV prompt emission, with a power-law decay of the flux with a $F \propto t^{-\alpha}$ behaviour, with α close to 1.

Some open questions regarding high-fluence GRBs and the light curve behaviour were also left behind, mainly limited by the small number of GRBs (only nine in the first Fermi Lat GRB catalogue) with an emission lasting for several hours after the prompt emission has finished.

The Second Fermi-LAT GRB Catalog (2FLGC) [42] reports 186 GRBs, 17 of which are short GRBs and 169 are long GRBs. Among the 186 GRBs observed, an emission above 100 MeV was found in 155 long GRBs and in 14 short GRBs.

High-energy prompt emission

As the jet composition, energy dissipation and particle acceleration mechanisms are not completely clear, there is a lack in the knowledge of the emission of gamma-rays from the prompt phase of the GRB. A fraction of the initial thermal energy of the fireball is radiated away at the photosphere, and at a larger radius internal shocks uses the kinetic energy of the jet to accelerate electrons which produce non-thermal gamma-rays via the synchrotron and inverse-Compton processes.

The prompt synchrotron emission in the internal shock model has some issues concerning the very low efficiency of the energy conversion, the lack of explanation of the observed spectra, only suggesting a way in which kinetic energy can be dissipated, and its predictive power concerning the time scales for the emission and the amount of energy that should be dissipated [54]. The GRB prompt emission can be produced by the synchrotron process provided that electrons are either continuously accelerated, or that there is some mechanism that prevents their rapid radiative cooling to ensure that the spectrum below the peak is consistent with observations [38].

In the context of the internal shock scenario, synchrotron self-Compton (SSC) models for the emission of GeV gamma-rays are a natural extension of the synchrotron model for the keV–MeV emission from GRBs. The temporal delay in the onset of the HE emission can be explained if inverse Compton scattering occurs in Klein-Nishina regime at early times, while at later times, conditions in the shocked region are such that the scatterings enter the Thomson regime. However an SSC component of internal origin cannot explain the flux excess which is sometimes observed below 50 keV [52].

An hadronic emission model could also be a possible explanation for the GeV emission, but the required energy would be two orders of magnitude larger with respect to the leptonic model (see previous section): energetic requirements can be relaxed by a very narrow jet with jet opening angle $< 1^\circ$ [52].

The observation of the prompt phase of GRBs at the temporal and energetic resolution of CTA will shed light on the physical processes at work in this early phase. This detection will most probably happen serendipitously in the field of view of the telescopes, larger with respect to existing IACT instruments. An observation of the prompt phase is unlikely to happen in a standard GRB observation due to the time needed by the telescopes to point the GRB direction and the short duration of the prompt phase.

High energy afterglow radiation

The highest energy reached by a photon coming from a GRB before the launch of EGRET was only 80 MeV and therefore EGRET was not design to detect such high energy photons so that its sensitivity to HE (above 100 MeV) was limited.²

In 1994, the event GRB940217 triggered both BATSE and EGRET (figure 2.11). This

²Since the spark chamber inside EGRET had a deadtime of about 600 ns to 100 ms after a trigger, it was impossible to measure the high flux of MeV gamma rays in a burst [55].

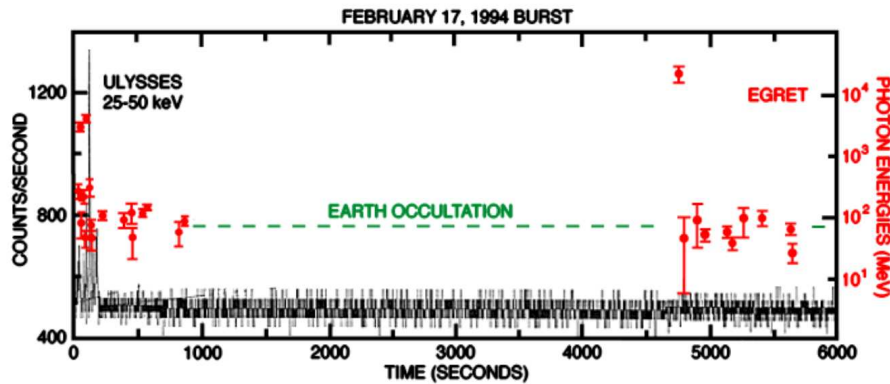


Figure 2.11: EGRET spark energies (red) as a function of time plotted together with the count rate of the Ulysses (BATSE) X-ray counter (black). At ~ 4700 seconds after the trigger we see the 18 GeV photon detected by EGRET spark chamber, after the satellite emerged from the earth limb (green). Credit: K. Hurley (UCB).

burst was significant because it lasted more than 90 minutes in the high energies, and in this time interval 18 photons were detected (with an expectation from the background of 4.7 photons). Even though the low energy emission was over (BATSE detection lasted only 180 seconds, measuring a fluence of $7 \cdot 10^{-4} \text{erg/cm}^2$), there was still a high energy emission taking place.

A really important discovery was the detection of a photon with an energy of 18 GeV approximately ~ 4700 seconds after the trigger, which raised the question of whether these sources had some breaks in their spectra, due to the presence of some additional components such as Inverse Compton. During its lifetime, EGRET found a delayed emission between the trigger of the GRB and the observation of the high energy emission for all the GRBs with hard spectrum [38].

As observed in the 1FLGC, the 2FLGC confirms that the high-energy emission recorded by LAT starts significantly later with respect to the low-emission recorder with GBM, with a huge spread in this time delay, going from almost zero to thousands of seconds. The emission is not only delayed at higher energies but also lasts longer with respect to the lower energies, the longest being GRB 160623A with a duration of 35 ks.

The delayed onset, the smooth light curve and the temporally extended emission above 100 MeV have led to the development of the external forward shock for being responsible of the high-energy emission during the early afterglow.

The specific flux from external forward shock can be specified with two characteristic frequencies, namely the synchrotron cooling frequency, ν_c , and the synchrotron peak frequency, ν_m . If we consider electrons accelerated and then cooled for a certain time due to the emission of synchrotron radiation, those with a Lorentz factor (LF) higher than a certain γ_c lose a most of their energy in this cooling time and their LF drops below this value: the synchrotron frequency $\nu_c \propto \gamma_c^2$ corresponding to this LF is the synchrotron cooling frequency. The second frequency, ν_m , is the typical synchrotron frequency of the accelerated electrons with the minimum LF. The specific flux at frequencies above $\max\{\nu_m, \nu_c\}$

is given by (eq. 81 from [38]):

$$f_\nu \propto E^{(p+2)/4} \epsilon_e^{p-1} \epsilon_B^{(p-2)/4} t_{\text{obs}}^{-(3p-2)/4} \nu^{-p/2} \quad (2.2)$$

where ϵ_e the fraction of energy density of shocked material given to electrons, ϵ_B the ratio between the energy density in the magnetic field and the energy density in the shocked circum burst medium (CBM) and the variable p is the index of the power law distribution for the injected electrons responsible for the emission. The spectral index and the time index of LAT lightcurve ($f_\nu(t) \propto \nu^{-1.1} t^{-1.3}$) satisfy almost perfectly equation (2.2) for synchrotron radiation from the shock heated CBM by the relativistic jet of a GRB when $\nu > \nu_c$. This relation is completely independent on the CMB density and its stratification and depends very little on ϵ_B .

These results lend strong support to the suggestion that high energy photons from GRBs detected by Fermi/LAT, for $t \gtrsim 30$ s, are produced via the synchrotron process in the external shocks. The detection of photons with energy greater than 10 GeV (with a rate of roughly 3 GRBs per year [42]) is anyway not easy to explain with the synchrotron interpretation (~ 50 MeV limit in the jet comoving frame for synchrotron emission) and not all the GRBs observed support this scenario. Some of them have a slower temporal decay, which would imply a continuous energy injection or some other sources, and some others instead have a light curve with a break or even hints of a plateau at late times [42].

Detection of GRBs at the highest energies The follow-up of GRBs performed by IACTs, such as the MAGIC and the H.E.S.S. experiments, has been conducted for many years with no clear detections until the beginning of 2019, when the MAGIC telescopes released an ‘‘Astronomer’s Telegram’’ reporting a >20 sigma significance for the detection of GRB 190114C in under 20 minutes of observation and at an energy greater than 300 GeV [56]. The observation, triggered by an alert sent from the Swift-BAT instrument, started just 50 seconds after the trigger and took place when the angle between the zenith direction and the source was already greater than 60 degrees and in the presence of partial Moon. This was the first detection of a GRB at VHE performed by an IACT system (recently published in [50]).

Another important result which also came in 2019 from the H.E.S.S. collaboration is the observation of two GRBs: the first one, GRB 180720B, has been detected with 5 sigma confidence level with the observation starting 10 hours after the observation,³ while the second one, GRB 190829A, had a redshift of 0.08 and has been detected with a >5 sigma confidence level with observations starting more than 4 hours after the trigger [58].

2.2.5 Progenitors

As already pointed out before, in 1993 Kouveliotou et al. [35] identified two phenomenological different classes of GRBs in the histogram of the durations of the GRBs (figure 2.6).

³Presented for the first time at the First CTA Symposium in May 2019 and later published in [57].

The short bursts had a duration of less than ~ 2 seconds and a hard spectrum and long ones are seen for more than ~ 2 seconds and have a softer spectrum. Approximately $3/4$ of the bursts are long and $1/4$ are short. This is referred to the BATSE sensibility range (30 keV–2 MeV), while in other detectors the fraction of short bursts is less than $1/4$, due to the fact that in different energy bands, the T_{90} time is different.

This distinction between short and long bursts, together with follow-up afterglow and host galaxy observations of GRBs, suggested that there might be two classes of progenitors for GRB: observations from BeppoSAX and Swift found that most long GRB host galaxies are found to be star-forming galaxies, suggesting that at least some long GRBs are associated with supernova Type Ic, coming from the collapse of massive stars. In 1998, GRB 980425 was seen by BeppoSAX in coincidence with the supernova SN 1998bw: this was the first evidence of a physical relationship existing between supernovae and GRBs.

For what concerns short GRBs, the Swift satellite found that some nearby short GRBs have host galaxies that have a low star formation rate (elliptical galaxies). Some short GRBs were found to be occurring in star-forming galaxies, but their locations indicate that they are located in a zone of the host galaxy with a low-star formation rate. All these arguments suggest that the type of progenitor for short GRBs are old objects such as two compact objects, either two neutron stars or a neutron star and a black hole, coming closer one another and at some point merging to form one final object¹.

Behind this simple classification, there is not a real agreement between the phenomenological classes and the physics-motivated classes: what is clear is that the observable quantities that are needed in order to classify the sources include supernova association, host galaxy properties, as well as the location within the host galaxy.

The confirmation of the origin of the short GRBs from a binary neutron star merger was confirmed in a recent observation: on the 17th of August 2017, the Advanced Ligo and Virgo detectors observed the gravitational-wave event GW170817 in coincidence with the short GRB (named GRB170917A) seen by the Fermi-GBM and by INTEGRAL [59]. This observation confirmed that at least some of the short GRBs are generated by the merging of binary neutron stars.

2.2.6 Prospects for CTA

Fermi-LAT has observed GRBs emitting in the GeV band for a total of $\sim 15 \text{ yr}^{-1}$ [42], revealing many interesting features of the GRBs observed at VHE: the emission extends up to 10–30 GeV, with the highest photons having an energy of ~ 100 GeV, it can be either associated with the prompt or the afterglow phase and lasting up to 1 day after the onset of the GRB, both for long and short GRBs.

While *Fermi*-LAT is limited above tens of GeV by the low photon statistics, IACTs can provide more information above those energies thanks to the larger collection area which brings a much higher photon statistics: however, the energy threshold and sensitivity of present IACTs limit the expected number of GRBs observed to be less than 1 per year,

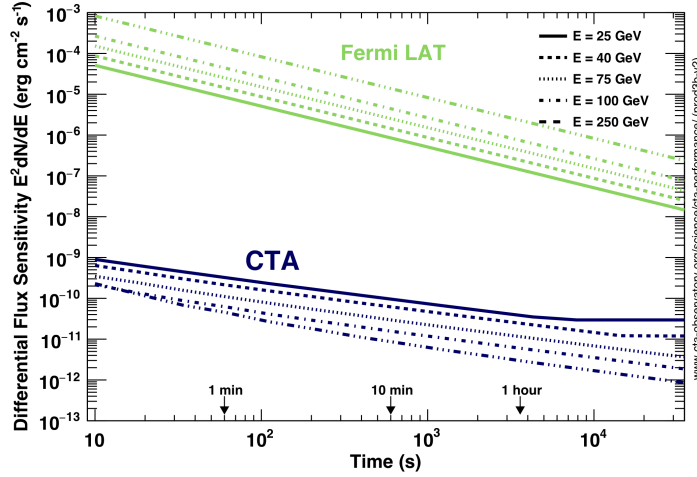


Figure 2.12: Differential flux sensitivity of CTA at selected energies as a function of the observing time in comparison with Fermi-LAT. This is the minimum flux needed to obtain a 5 sigma detection from a point-like gamma-ray source, with 5 bins per energy decade, and a minimum of 10 excess photons is required. Image from [7].

with no GRB seen from any IACT until the beginning of 2019. Due to the larger number of telescopes, the low energy threshold and the increased sensitivity, CTA is expected to reach a detection rate of more or less a GRB per year, being orders of magnitudes better than Fermi in the observation of transient sources (see figure 2.12).

The need for a low repositioning time after an external trigger needing low observations at $E > 20$ GeV, has driven the requirements for the telescopes. The Large Size Telescopes (LST) are required to point to any direction in the sky within 30 seconds from the received alert, while the Medium Size Telescopes and the Small Size Telescopes have a requirement of 90 seconds [7]. The observation of extragalactic transient sources, due to their rapidly fading flux coupled with the absorption of the high energy photons due to the EBL, is suited to the LSTs, that have both a fast re-positioning time and the best sensitivity at the lowest energies. Prompt emission from GRBs at high redshift might also be observable, provided that they have a flux which is sufficiently high (prospects for the observation of GRB 080916C, the GRB with the highest redshift detected by Fermi, are reported in [60]).

Due to the larger field of view with respect to existing IACTs, coupled together with new observing methods like the divergent pointing mode, CTA will have a larger probability of detecting a GRB while in sky survey mode: if we refer to the MSTs, the probability of a serendipitous detection in the field of view is 0.08 yr^{-1} , while if the survey is done using a divergent pointing mode, this number can be as high as few GRBs per year (sec. 8.1.1 of [24]). A serendipitous detection in the field of view would allow CTA to observe the prompt phase of GRBs and to send an alert to other telescopes and satellite: CTA will be able not only to receive alerts and re-point accordingly, but also to send alerts. Such detection would be really interesting not only to understand the physics of GRBs in the early phases, but also to test fundamental physics such as Lorentz invariance violation [60].

Observation strategy

Once an alert has arrived, it will be followed if the source is at zenith angle lower than 70° , during dark time (no moon) and for a 2 hours snapshot: the LSTs will be repointed as soon as possible to guarantee the maximum sensitivity at the lowest energies, with the MSTs being slightly slower and arriving within one minute from the LST. The alert rate is expected to be around 12 h/site/year and the total estimated observing time is 50 h/yr/site [24]. If the Real-Time Analysis (RTA) detects some signal from a GRB, the alert will be followed for as long as the target is observable. If the GRB is not immediately visible (due to sun presence or source not yet visible), it will be important to repoint the telescopes as soon as the source becomes visible, i.e. its altitude above the horizon is greater than 20° or the sunset has passed.

It will be even more important to follow a GRB alert with partial moon condition, which is the observing condition under which MAGIC has performed the first detection of a GRB by an IACT [56]: the observation under partial moon light might increase also by 50% the probability of detecting such sources [24].

Together with a fast repointing of the CTA telescopes, the synergies with the optical and infrared telescopes will be very important for the determination of total burst energy, the jet collimation angle and the ambient density, with the most important parameter obtained in the optical band being the redshift of the source. It might be desirable to have an on-site optical telescope dedicated for CTA follow-up which can localise a majority of the afterglows to a sufficient level of accuracy, so that larger telescopes can be alerted for the follow-up of the event.

CTA will work together with other observatories, telescopes and satellites in the *Astrophysical Multi-messenger Observatory Network* (AMON), in order to search for sub-threshold signals in coincidence between all the observatories in a multi-wavelength and multi-messenger scenario.

CTA performances for GRBs detection

In order to estimate the performances of CTA on the observation of GRBs, simulations have been carried using as test models some Fermi-LAT GRBs, whose spectra is well described by a power-law, extrapolating the spectra at higher energies and taking into account the EBL absorption.

Thanks to the great sensitivity of CTA, it will be possible to obtain high resolution light curves of the brightest GRBs, which will provide a deeper knowledge on the central engine of the GRB, the formation of the jet and the emission mechanism, to discriminate between hadronic or leptonic emission processes, as described in the previous sections. Moreover, while the cut-off on the EBL is time-independent (see appendix A), there might be some other cut-off in the spectra at VHE with a time-dependent behaviour which might show new physical processes happening.

The detection rate for GRBs seen by CTA is of the order of one per year per site, with those estimates being carried from the present population of GRBs seen by Fermi-LAT,

which might underestimate the number of GRBs that CTA might be able to see: there might be GRBs with lower luminosities that haven't already been discovered, but that CTA would be able to detect, even though those phenomena are still unknown and it's therefore not possible to get an estimation on their detection rate.

The details on the work done for the Consortium paper on the detection rate of GRBs with CTA will be covered in section 4.

2.2.7 Summary

So, to summarize, these are some of the important concepts regarding GRBs:

1. They occur at cosmological distances, typically at redshifts of a few.
2. They are generated by outflows with ultrarelativistic bulk velocities.
3. Their prompt emission is accompanied by afterglows that span from the radio up to the VHE bands and gradually decay over hours to days or more, most likely emitted by high-energy electrons accelerated in the blastwave resulting from the interaction of the outflow with the ambient medium (figure 2.8).
4. Those with durations longer than ~ 2 s (“long” GRBs) exhibit properties systematically different from those with shorter durations (“short” GRBs).
5. At least some long GRBs are associated with the core-collapse supernova events of massive stars and some short GRBs are associated with the merging of two neutron stars (as seen from recent observations).
6. The emission at the VHE in the afterglow from the inverse Compton up-scattering of synchrotron photons by high-energy electron has been observed by MAGIC in GRB 190114C [61] and two more have been observed by H.E.S.S., namely 180720B (5 sigma detection 10 hours after the onset) and 190829A (at a redshift of 0.08) . More observations of GRBs at very high energies are needed in order to put constraints on the emission mechanisms, discriminating between hadronic or leptonic models [50].

2.3 Gravitational Waves counterparts

2.3.1 Introduction

Considered for a long time a proof of the general theory of relativity published by Albert Einstein at the beginning of the XX century, the search for a direct emission of gravitational waves has been going on for several decades with different experiment, such as resonating antennas, so called Weber bars, or interferometers. Decades-long observations of the rotational period of the binary pulsar B1913+16, the Hulse–Taylor binary, has shown that the shift of the periastron time that the system has gone through in the last decades is perfectly compatible with the emission of gravitational waves from the system



Figure 2.13: The Virgo interferometer (from The Virgo collaboration).

as predicted by the general relativity [62]: this has been for many decades a strong indirect proof for the existence of gravitational waves, even though the direct observation of these phenomena was missing.

A great breakthrough came in 2015 with the first observation of a gravitational waves signal by the two LIGO interferometers of the LIGO-Virgo Collaboration, a observation thanks to which Rainer Weiss, Barry C. Barish and Kip S. Thorne got the Nobel Prize in Physics in 2017. The two LIGO interferometers, one located in Hanford (Washington, USA) and the other one in Livingston (Louisiana, USA), detected a very short signal (0.2 seconds long) compatible with the emission of gravitational waves from a merger of two black holes with masses of 35_{-3}^{+5} and 30_{-4}^{+3} solar masses, resulting in a post-merger mass of 62_{-3}^{+4} solar masses: the missing 3 ± 0.5 solar masses has been radiated away in the form of gravitational waves [63].

The search of gravitational waves signals has continued since then and there have been 10 black hole merger events reported in the first and the second observation runs [64], named O1 and O2: the observation run O3 is presently going on and many other similar events have been observed.

Together with the black hole mergers, during the O2 observation run of the LIGO-Virgo interferometers, a binary neutron star merger event was seen on the 17th of August 2017 in coincidence with a GRB seen from Fermi and INTEGRAL [59]: the signal lasted roughly 100 seconds and was compatible with the merging of two neutron stars, while the peak of the gravitational waves emission reached us 1.74 seconds before the GRB was

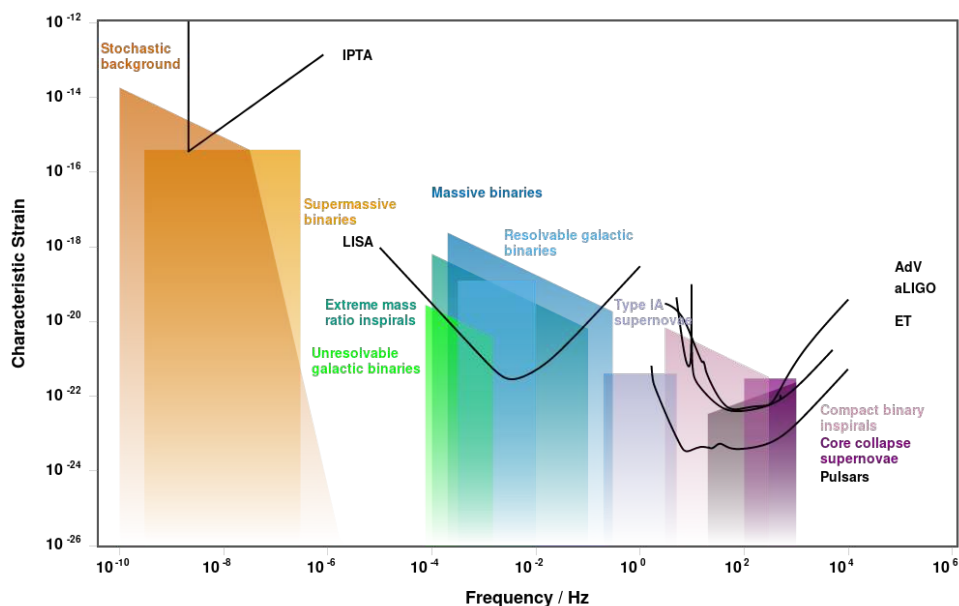


Figure 2.14: Sources of gravitational waves and the detectors. Plot created at <http://gwplotter.com/>.

recorded from the Fermi-GBM instrument. This observation was the confirmation that short GRBs, at least some of them, are originated by the merging of two binary neutron stars.

2.3.2 Physics of GWs

Gravitational waves are perturbations of the space-time that can be emitted by different types of objects in the Universe, with the strongest signals being those emitted by more massive and less spherical systems. Those ripples in space-time can be detected directly if we can measure the stretching and squeezing of the specific length that we are taking under consideration. These small deformations are measured either with the Weber bars, aluminium bars suspended and cooled to almost 0 K (that have not produced any signal of gravitational wave yet), or laser interferometers, using a laser beam which travels inside vacuum tubes for several kilometres and it is used to measure the length of the interferometer's arms to see if it has changed due to the passage of a gravitational wave.

In his paper “On Gravitational Waves”⁴, Einstein wrote that “...a mechanical system which permanently retains spherical symmetry cannot radiate...”, so that, for example, a massive star reaching the end of its life and exploding as a supernova uniformly in all directions, without any preferred direction being affected by the explosion, will not emit gravitational waves. A perfect spherical symmetry, with no emission of gravitational waves, is very unlikely to happen in nature but this gives a hint to the types of sources to look for with interferometers and other instruments to detect gravitational waves, namely objects that are not spherically symmetric and also massive. Figure 2.14 represents a list

⁴Preussische Akademie der Wissenschaften, Sitzungsberichte, 1918 (part 1), 154–167. <https://einsteinpapers.press.princeton.edu/vol7-trans/39>

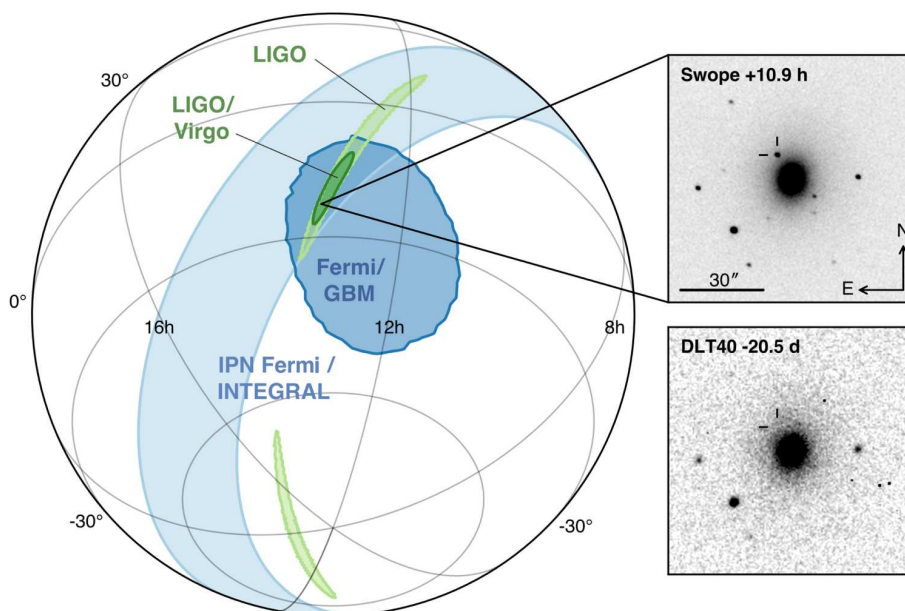


Figure 2.15: *Left panel:* localization of the gravitational wave, gamma-ray and optical signals for the event GW170817. The light green is the 90% probability region from LIGO (190 deg²) and the dark green is the initial LIGO-Virgo localization (31 deg²). Light blue is obtained from the time delay between Fermi and INTEGRAL, whereas the dark blue corresponds to the Fermi-GBM localization. *Right panel:* optical signal from the host galaxy NGC 4993 20 days before and 10 hours after the gravitational wave signal. Image from [65].

of sources that are expected to emit gravitational waves together with the sensitivities of present and future experiments.

The two types of sources seen by Ligo and Virgo so far have been binary black holes and binary neutron stars mergers: these are very massive objects that can come very close one another and therefore the associated emission of gravitational waves is much larger with respect to other coupled systems, happening at larger distances and involving smaller masses.

Electromagnetic follow-up

When a gravitational wave event is observed by the interferometers, a fast alert with an associated probability skymap is sent to other telescopes, observatories and satellites, which then start the observation to look for an emission in other wavelengths in the region of the gravitational wave signal. Together with the probability skymap, a fast analysis is carried in order to understand which type of event has been detected.

The two classes of events detected by the LIGO and Virgo interferometers has been, as already said, a binary black hole merger and a binary neutron star merger. The first class of events is not expected to emit electromagnetic radiation and an observation of electromagnetic radiation in coincidence with a gravitational wave event associated with this source class would favour new emission models [66]. The second class of events is expected to emit electromagnetic radiation in all wavelengths (as observed for the event

GW 170817 [65]) and it's a mandatory class of events to observe with as many instruments as possible.

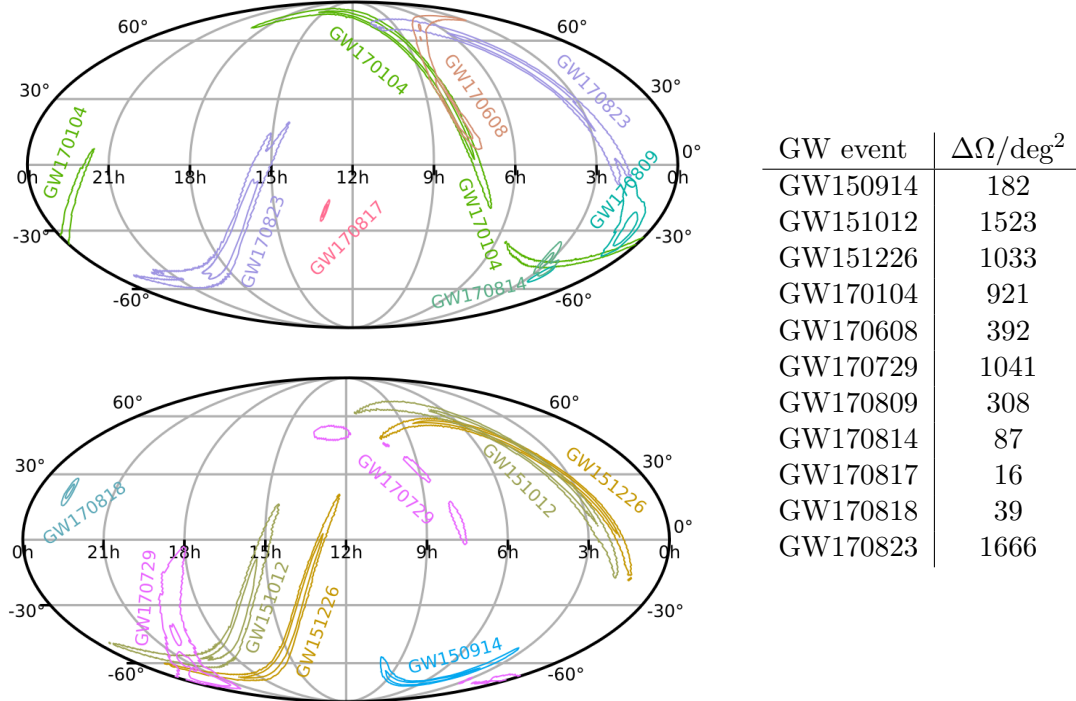


Figure 2.16: The contour on the left figures shows the 50% and 90% credible regions for the sky localisations of all GW events in O1 and O2 (values reported on the table on the right). In the top figure are reported the GW events for the O2 run (GW170817, GW170104, GW170823, GW170608, GW170809, and GW170814) for which the alerts were sent to the EM observers. The bottom figure shows the GW events seen in O1 (GW150914, GW151226, and GW151012), along with O2 events (GW170729 and GW170818) not previously released to EM observers (figures and data from [64]).

Since the interferometers are not giving a pin-point location in sky to look at but a more or less big sky map (see figure 2.16), it's challenging for other instruments to perform a follow-up of the event, due to the non-precise sky localization of the source position and the limited field of view of such instruments.

Due to the observation of VHE emission from a long GRB and the association of short GRBs with the emission of gravitational waves, it will be important to follow-up these events with CTA and the other IACTs. The method followed by IACTs uses a series of pointings with variable duration in order to properly cover the region to observe (see for example MAGIC and H.E.S.S. strategies in [67] and [68]), making sure to cover the largest possible area of the probability sky map and to have each pointing last long enough to make sure that if the source is in the field of view, it's not missed due to a too short exposure time.

2.3.3 Prospects for CTA

Since it's possible that mergers of binary black holes and neutrons stars (or mixed systems) might accelerate particles to TeV energies, CTA will perform follow-up of the GW alerts

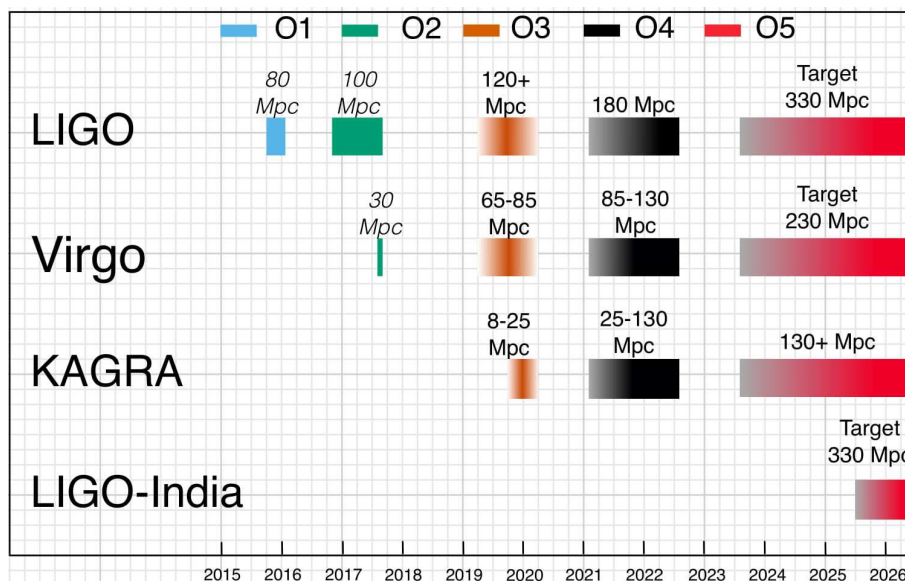


Figure 2.17: Future gravitational waves detector and their average reach to binary neutron star mergers (figure from [70]).

sent by LIGO and Virgo. These events are presently being followed at the highest energies, with no hint of detection so far by any IACT system during the last observation runs (see [67], [68] and [69]). Event GW170817 in figure 2.15 was followed for many months in all wavelengths, from radio to gamma, with no signal detected by any IACT at VHE [65]. In the scheme of the CTA Key Science Project on transient sources, gravitational wave transients are ranked as one of the highest priority to be studied, with a proposed observation plan of 20 h/year/site for the early phase and 5–10 h/year/site afterwards [24].

The localisation errors of GW events reported in the O1 and O2 observation runs of LIGO and Virgo showed a large variation depending on the event, ranging from 16 deg² to 1666 deg² (see figure 2.16) with few events having small localisation errors. In the next years more events are expected to be triggered, both because LIGO and Virgo will reach their best sensitivity which means that they will be able to observe a bigger volume in space, and because there will be more interferometers looking for gravitational waves, namely the Japanese KAGRA⁵ and the foreseen LIGO-India. The plans for the next decade of gravitational waves interferometers is reported in figure 2.17.

Even though there will be more interferometers looking for some gravitational waves emission, since in the first years of operations their horizon will be limited, the localization errors are not expected to improve a lot, remaining in the 100–1000 deg² range [24]: with an improvement in the GW horizon in the future observing runs, there will be many more events triggered, with a huge number of events happening at the sensitivity limit and therefore associated with large localisation errors. The observation runs will observe events with a relatively large and asymmetric localization errors until the advent of the

⁵The KAGRA Collaboration has signed an agreement with the LIGO and Virgo Collaborations to take part in the O3 observation run from December 2019.

third generation of detector, such as the Einstein Telescope⁶. This new experiment will be a triangular interferometer with 10 km long arms, located between 100 m and 300 m below the ground level, with an expected sensitivity one order of magnitude better with respect to existing interferometers.

CTA will have a great advantage with respect to other instruments and wavebands for what concerns the follow-up of GW alerts, due to the large field of view of the telescopes, the large number of telescopes, the fast response time and also a divergent pointing mode (see section 7.1). The follow-up of gravitational waves alerts with CTA may offer better localization with a more efficient search over a large area in the sky and the detection of some signal in coincidence with the GW alert would provide further informations on the source and its characteristics in a multi-wavelength and multi-messenger observation.

The observation of VHE photons, which might also not come from the jet pointing towards us but from the interaction of an off-axis jet with the ambient media, would provide unique informations for what concerns the dynamics of those events, together with observations in other wavelengths such as optical and infrared [24].

More details on the work done for the Consortium paper on the GW follow-up program with CTA will be covered in section 4 (GW pipeline for Consortium Paper).

⁶<http://www.et-gw.eu>

Chapter 3

The First Data Challenge

The focus of the First CTA Data Challenge (DC-1) was to validate the core versions of the science tools that are being developed for the high-level analysis of the data coming from the CTA telescopes: the two main tools are `ctools` [71] and `gammapy` [72]. The data for DC-1 came from the simulation of a physical model of the sky which was provided by different working groups, whereas as soon as the telescopes will start taking data, the future data challenges will be done using real data.

At the moment of writing the DC-1 is over and a close-out document¹ is being finalized: this includes a complete description of all the physical models included in the DC-1 and the outcome of the analyses that has been done, together with suggestions and improvements for the following data challenges.

In this chapter I will present an outline of the close-out document with a particular focus on my personal contribution to the DC-1 for the creation of the extragalactic sky model.

3.1 Goals

The DC-1 had many goals, which can be summarised as follows:

- Increase the number of people analysing high-level data within CTA;
- Prepare for the analysis of the first real CTA data;
- Validate current data formats and find missing informations or specifications;
- Validate features and completeness of science tools packages;
- Push the development of new algorithms;
- Ensure the compatibility of CTA data with multiple tools;
- Get semi-realistic assessment of Key Science Projects (KSPs).

¹https://forge.in2p3.fr/attachments/download/62211/CTA_DC1_CloseOut.pdf. The URL is not of public domain and links to an internal CTA webpage.

3.2 Technical inputs

In order to simulate the physical model of the sky, many inputs from different working groups were needed, from the subgroups in the Physics (PHYS) working group to the Analysis and Simulation Working Group (ASWG).

3.2.1 Science tools

The two science tools used for the production and the analysis of DC-1 data are `ctools`² and `gammapy`³.

`Ctools` is a software package for the analysis of CTA data (the analysis of the H.E.S.S. data is also supported, together with Fermi and COMPTEL data) which works on top of an external library called `gammalib`⁴: these libraries are written in C++ with wrappers for Python which make `gammalib` and `ctools` easy-to-use in a Python script. The tools provided in `ctools` are similar with respect to the one adopted by the Fermi satellite, so that a user familiar with the Fermi analysis can easily perform an analysis using `ctools`. The `ctools` version used in the DC-1 was the 1.5.0, while the most up-to-date `ctools` version at the moment of writing is 1.6.3.

`Gammapy`, is a Python package for gamma-ray astronomy which is mainly built on `Numpy`⁵ and `Astropy`⁶, two libraries that are heavily used in the scientific community of Python users. The idea behind `gammapy` is to use the functionalities that are developed in external Python libraries, without re-implementing already existing algorithm and functionalities.

The simulated data have been generated using `ctools` and could then be analysed using both `ctools` and `gammapy`.

3.2.2 IRFs

The ASWG provided the IRFs (see section 1.3) to be used by the analysis tools to produce and analyse the simulated data: these IRFs correspond to an ideal CTA with good and stable atmospheric and instrumental conditions.

The IRFs used for the DC-1 were produced with cuts and parameters in the IRF-making chain that provide the best sensitivity when observing a source with a spectrum similar with respect to the Crab Nebula (same slope but different scaling factor applied to the spectrum) for a cumulative time of 50 hours: in particular 4 IRFs have been used, 2 per each site (North and South) and 2 per each zenith angle, namely 20° and 40°.

Since these IRFs are optimized for sensitivity studies, they would give non-optimal results when applied to dark matter searches, in which the search for lines in the spectra should be carried out with an IRF that has been optimized for energy resolution.

²<http://cta.irap.omp.eu/ctools/>

³<https://gammapy.org/>

⁴<http://gammalib.sourceforge.net/>

⁵<http://www.numpy.org/>

⁶<http://www.astropy.org/>

3.2.3 Physical models

As already said previously, the DC-1 was not intended to be used to derive detection rates or perform deep studies concerning what CTA will be able to see it, but was more focused in testing analysis tools and getting more people involved in the analysis of CTA data. Therefore, the physical models provided by the different working groups were simple models in which many important effects might not have been taken in account.

The format used to describe the models is an XML file, which is the format used by ctools: the model for each source is split in a spectral component, a spatial component and a temporal one (if needed) and many different options are available for each of these components.

The sources in the sky model of the DC-1 where:

Known bright sources: the sources for the GPS sky model have been chosen from gamma-cat⁷. The energy interval used for the models is 100 GeV – 100 TeV, the sources having a galactic latitude b such that $|b| < 5^\circ$;

Synthetic population of PWNe: Pulsar Wind Nebulae were modelled in the DC-1 in a phenomenological approach, extrapolating the logN-logS distribution obtained by all the known established and candidate PWN in the Galactic Plane Survey made by the H.E.S.S. collaboration. A PWN is a type of nebula located around a pulsar which stores and displays the radiative output of the pulsar during tens of kiloyears: the PWN is energised by the particle flux from the pulsar and cooled by radiative, adiabatic and escape losses [73]. The total number of simulated PWN is 650, with fluxes as low as 0.05% the Crab nebula flux and a log-parabola spectrum, with parameters chosen from the distribution of known PWNe, and a Gaussian spatial template with no energy-dependent morphology.

Synthetic population of SNRs: only young SuperNova Remnants were included in the DC-1 sky model (from [74]).

Gamma-ray Binaries: 5 known TeV-emitting binaries were included (LS 5039, LS I+61 303, PSR B1259-63, 1FGL J1018.6-5856 and HESS J0632+057). For these, the lightcurve has been modelled and spectra are given as power laws (except one which has a high energy break). For this data challenge, super-orbital variability, orbit-to-orbit variability and low energy breaks are neglected.

Bright pulsars from Fermi-LAT: the 12 most energetic gamma-ray pulsars seen by Fermi are included in the DC-1, which are those with a significant pulsed emission above 25 GeV. These sources are selected from the 1FHL which is a catalogue of 514 sources made by the Fermi-LAT Collaboration with data above 50 GeV collected in the first three years of the satellite [75]. The spectra are modelled as power-laws with

⁷<https://gamma-cat.readthedocs.io>

indexes and normalizations obtained from the fit of the Fermi-LAT data at energies above 10 GeV and extrapolated to VHE without including a cut-off. The shapes of the pulse profiles are determined folding 5 years of Fermi-LAT data above 10 GeV.

Interstellar radiation: there’s no population of unresolved sources. The model includes predictions from CR propagations code for Inverse-Compton emission and gas-related emission (bremsstrahlung and π^0 decay), with an angular resolution for the latter improved using Planck dust maps.

Fermi bubbles: the Fermi bubbles, extended structures located above and below the galactic plane emitting gamma rays discovered by Fermi, were also included in the DC-1. The sky model is based on lower-energy measurements by Fermi-LAT and the morphology of the bubbles was modelled based on the results of the spectral component analysis in Fermi-LAT, and their spectrum following the log-parabolic fit to the data above latitudes of 10 deg.

Dark Matter: Dark Matter is modelled at the center of the Galaxy and in other few selected dwarf galaxies.

Active Galactic Nuclei: there are more than 500 sources simulated for the AGN monitoring program and the Extragalactic Survey. I worked specifically in the simulations of the the AGNs and provided the XML models for such simulations. I will present those models in detail in the next section.

The energy range used for all the simulations is 30 GeV–160 TeV, a part for the AGN data for which the energy range is 30 GeV–50 TeV.

3.2.4 The extragalactic sky

I contributed to the DC-1 for what concerns the creation of the extragalactic sky model. An important role when treating sources at high redshift, such as AGNs, is played by the EBL (see Appendix A). The approach followed for the creation of the extragalactic sky was to include the highest number of sources using informations from VHE catalogues, so that the absorption of the EBL is already taken into account. In order to include sources with low-energy spectra, that are assumed to be not influenced by the EBL, their spectra have been extrapolated to the VHE absorbing the flux according to the EBL model from Franceschini [76]. For a more detailed modelling of these source, the spectra should have been deabsorbed for the EBL and then reabsorbed for it over the large CTA energy range.

The first catalogue used is a catalogue of sources observed by ground-based Cherenkov telescopes, namely the TeGeV Catalogue⁸ collected by the ASI Space Science Data Center (SSDC).

In order to get the list of sources, I applied some cuts to this catalogue, so that the galactic sources were removed, together with sources tagged as unidentified (“UNID”)

⁸<https://www.ssdsc.asi.it/tgevcatalog/>

and with a galactic latitude between -10° and 10° and sources with no complete data for spectra characterization: out of the 155 sources available at the time of the creation of the models⁹, 46 have been extracted from this catalog.

The second catalogue used is the 2FHL [77], which is a catalogue of 360 sources made by the Fermi-LAT Collaboration with data above 50 GeV collected in 80 months. The data can be either downloaded from the FSSC website¹⁰ or from the ASI SSDC website¹¹. I downloaded the catalogue from the SSDC webpage, and performed a source selection similar to the one used for the TeGeV, excluding galactic source, unidentified sources on the galactic plane and already selected sources from the TeGeV. Out of the 360 sources in the 2FHL, 136 new sources were selected, 11 of which were also present in the TeGeV but didn't have data in that catalogue, 87 were also present in the 1FHL but with no redshift in that catalogue, plus other 38 2FHL sources: at the end of the selection, 231 sources from the 2FHL were included in the source selection. Similarly with respect to what was done for the TeGeV catalogue, the spectra of the sources were considered as already absorbed by the EBL since the 2FHL is a VHE catalogue with data above 50 GeV, therefore the spectra were created with no need to absorb for the EBL, only extrapolating the source to higher energies.

In order to include more sources, I also considered some sources in the 1FHL [75] also downloaded from the SSDC webpage¹². Since the spectra of the sources in the 1FHL starts from 10 GeV, the spectra is weakly affected by the EBL and in order to extrapolate the flux at higher energies I had to take into account only those sources in the 1FHL for which the redshift had been measured, so that the spectra could be absorbed according to it. As previously done for the TeGeV and the 2FHL, the galactic sources and the unidentified sources close to the galactic plane, together with those already included, were excluded from the selection, leaving 184 sources at this stage of the selection.

In order to extend the population of sources to lower fluxes and to other types of AGNs, other catalogues were used to extract the names of the sources to be included, whereas their spectral data and names, were taken from the 3FGL, which is a catalogue of 3034 sources detected in the first four years of observations of Fermi-LAT in the energy range between 100 MeV and 300 GeV (downloaded from the SSDC webpage¹³). Then I extracted a list of sources from the 3FGL among the thousands of sources available. Here is a list of the selected sources:

- 60 FSRQ: Flat Spectrum Radio Quasars with a redshift lower than 1.4, both ISP (“Intermediate Synchrotron Peak”, with $10^{14.5} < \nu_{\text{peak}}^s < 10^{16.5}$ Hz) and LSP (“Low Synchrotron Peak”, with $\nu_{\text{peak}}^s < 10^{14.5}$ Hz). We decided to cut at this value of redshift since the maximum redshift at which an AGN has been seen so far by

⁹The catalogue used was the second version, whereas the more recent one which can be found in the SSDC website is the version 4 released in December 2018

¹⁰<https://fermi.gsfc.nasa.gov/ssc/data/access/lat/2FHL/>

¹¹<https://www.ssdsc.asi.it/fermi2fhl/>

¹²<https://www.ssdsc.asi.it/fermi1fhl/>

¹³<http://www.ssdsc.asi.it/fermi3fgl/>

IACTs is 0.9, and therefore the extension to 1.4 seemed a reasonable value to include a large population of sources. The value of the synchrotron peak was taken from the 3FHL, which is a catalogue reporting the locations and spectra of sources detected between 10 GeV and 2 TeV during the first 7 years of the Fermi mission using the Pass 8 event-level analysis.

- 60 HSP BL Lac: “High Synchrotron Peak” BL Lac ($\nu_{\text{peak}}^s > 10^{16.5}$ Hz). Those sources were selected using the 2WHSP catalogue, which is a catalogue of HSP blazars. In order to obtain the redshift of as many sources as possible, the 3FHL and the 5BZCAT catalogues were also used, the latter being the 5th edition of the Roma-BZCAT, a multi-frequency catalogue of Blazars, with data from the radio band to the high energy band. Sources with a lower limit for the redshift were put at that redshift, whereas the average redshift of the 327 BL Lac sources from the 3LAC (the third catalogue of AGN detected by Fermi-LAT between 100 MeV and 300 GeV) was assigned to sources without redshift, namely 0.4.
- 24 ISP BL Lac: “Intermediate Synchrotron Peak” BL Lac with index in the 3FGL lower than 2.25 and with spectra described by power laws and not Log parabola.
- 23 LSP BL Lac: “Low Synchrotron Peak” BL Lac with index in 3FGL lower than 2.25 and with spectra described by power laws and not Log parabola. The redshift of these sources is less than 1.4, as for previous sources.
- 2 NLSy1: Narrow line Seyfert I. The 2 NLSy1 with the lower redshift were selected for the DC-1, namely 3FGL J0325.2+3410 and FBQS J1644+2619; the second one is not in the Fermi catalogues but the detection was reported in [78]:
- 6 RDG: Radio Galaxies with index in 3FGL lower than 2.2 and a variability index in 3FGL greater than 20.
- 2 SGB: StarBurst Galaxies.

The LogN-LogS histogram for all the selected sources is reported in figure 3.1 (left panel), together with the LogN-LogS for some selected sources (right panel). Since the Extragalactic survey will cover the entire sky but only a fraction of it, the right panel in figure 3.1 shows the LogN-LogS for those sources that have been detected from the simulation of this region of the sky.

In order to have some variable sources, particularly important for the AGN monitoring program KSP in which the AGNs are constantly monitored looking for flares, a variability was added to 35 sources as a multiplicative factor to the overall flux (the “shape” of the spectra is kept constant and it is not varying with time). The light curves were collected from the list of monitored sources by Fermi-LAT¹⁴, which are sources that have been

¹⁴https://fermi.gsfc.nasa.gov/ssc/data/access/lat/mssl_lc/

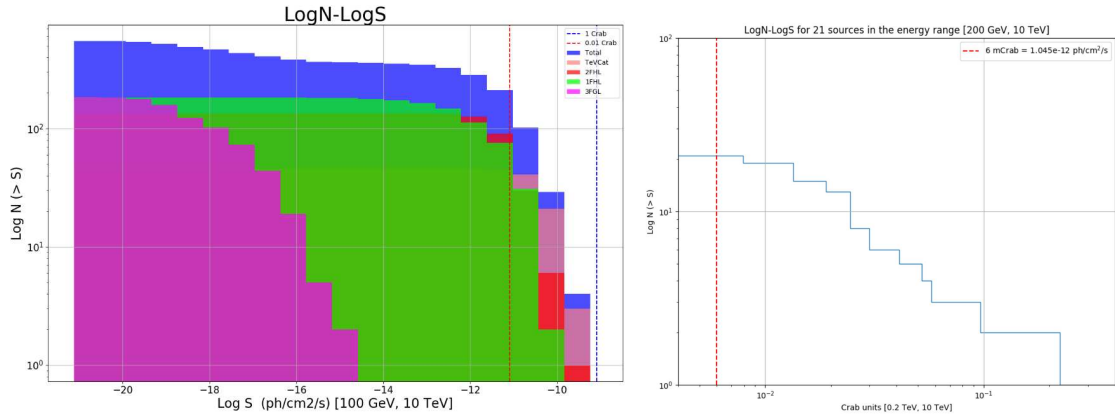


Figure 3.1: LogN vs LogS for the sources in the extragalactic model. The plot shows, as a function of the flux, the number of sources with a flux greater than a certain value. The right plot is the LogN-logS for the full population of sources, highlighting the contribution coming from each catalogue, whereas the plot on the right is a zoom in a slightly different energy range of the highest flux interval.

reported to have a flux while flaring greater than the threshold value of $10^{-6}\text{cm}^{-2}\text{s}^{-1}$: if a source overcome this value, it's added to the list of monitored sources.

The assumption for the modelling of the variability of these AGNs is that the source varies by the same amount with respect to the steady flux both in the Fermi energy range and in the CTA one. Even though this is a strong assumption, the goal of this work was only to have some variable sources, without a detailed modelling of the variability of these sources in the VHE regime.

In order to properly assign the fits models for the variability to the right sources, we selected a list of variable sources in the 3FGL, found their “Counterpart name” in the 3FGL and use this name to search for the source in the Fermi light curve monitoring webpage: e.g., the source 3FGL J1256.1-0547 is called 3C 279 (from the third Cambridge catalogue of radio sources), which is the name used in the Fermi light curve monitoring webpage. For each source I took the weekly light curve, in which each data point is the average flux in one week, and created a fits file with time and normalization: since ctools allows only to have a normalization between 0 and 1, the overall curve was normalized to 1 with a normalization coefficient in the ctools xml model used to scale the values in the fits file to the correct ones. The points in the light curve which were reported as upper limits have been set equal to the catalogue flux (see figure 3.2 for Markarian 421). The flux variability happening on timescales shorter than a week is averaged in this approach.

The outcome of this selection was a list of sources with spectra measured at VHE for which it was not necessary to absorb the spectra due to the EBL, and a list of sources in the 1FHL and 3FGL for which the spectra was absorbed taking into account the redshift.

While in the first case the spectra were simply specified as power laws (with a normalization factor, an index and a pivot energy needed to specify the spectrum), in the second case in order to take into account properly the EBL across the whole energy range, the spectra were provided as a separate file for each source (energy in one column and flux in $ph/cm/s/MeV$ in the other column); the temporal behaviour was specified with a FITS

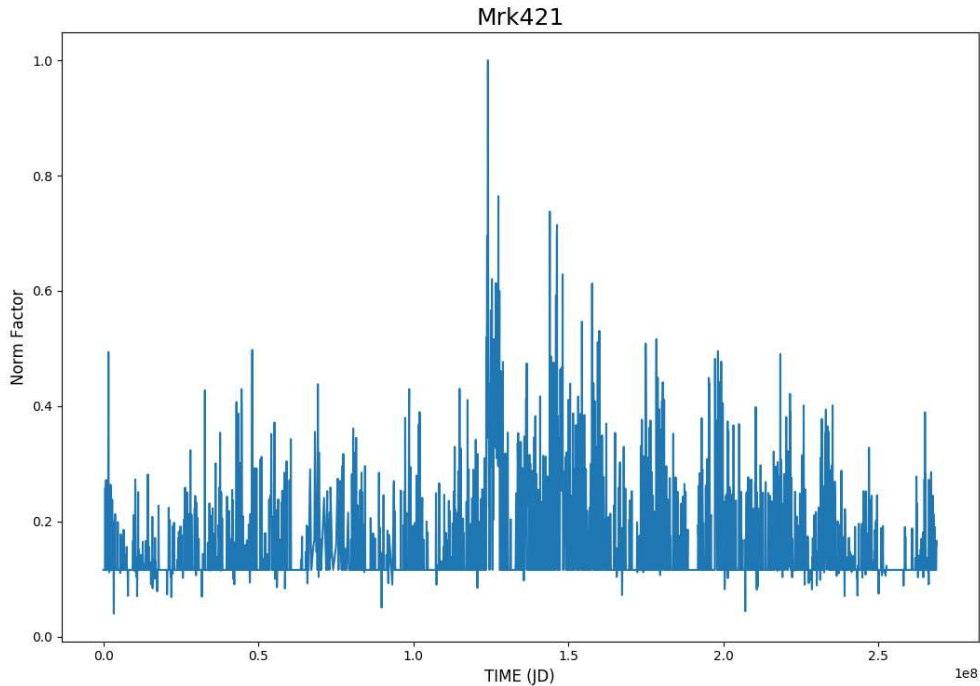


Figure 3.2: Light curve for the source Mrk 421. The maximum of this plot is one because a variable model in `ctools` requires the fits file to have a normalization factor between 0 and 1. Everything can be scaled with an external parameter.

file with the time in the first column and the normalization factor in the second one.

At the end of this selection process I had a text file with one source per line (551 sources in total), each having the spatial, spectral and (if present) temporal parameter.

Since I had to create the XML models both for the GRB simulations and for the DC-1, I built a Python script to create those XML models in an easy way. The script can build almost all the different spatial, spectral and temporal models that can be read by `ctools` and the only input needed is a text file with all the characteristics of the source specified according to some rule regarding naming and number of input parameters. For example, this string:

```
Src_name Point 1 329.719 -30.2217 0 0 0 FUNC 1.0 path/spec.out 2.0 path/lc.fits
```

is used to generate a model of a **Point** source named **Src_name**, for which we will calculate the TS value during the likelihood computation (parameter equal to 1 after **Point**), located at $RA = 329.719$ and $DEC = -30.2217$, for which the spectrum is a file (**FUNC** label) with a normalization of 1.0 and the path of the file is **path/spec.out**. The temporal model has a normalization of 2.0 and the path of the fits file to describe the time evolution is **path/lc.fits**.

This script was then completed and later used for a tutorial session that I prepared together with other colleagues for the Sexten Center for Astrophysics during the school “Gamma ray astrophysics with CTA”¹⁵. The tutorial sessions¹⁶ was given to students

¹⁵<https://www.sexten-cfa.eu/event/gamma-ray-astrophysics-with-cta/>

¹⁶The material can be found at https://github.com/HESOFTS/sexten_2017.

KSP	Duration	Pointings	Simulated dates
Galactic Plane Survey	1620 h (1020 S, 600 N)	3270	2021-01-01 - 2021-04-18
Galactic Centre Survey	825 h (S)	1671	2021-01-01 - 2021-03-29
Extragalactic Survey	500 h (200 S, 300 N)	1271	2021-01-01 - 2021-02-25
AGN monitoring	960 h (N)	1920	2021-01-01 - 2022-07-09

Table 3.1: Distribution of the observing times in hours between the different KSPs for the North array (N) and the South array (S). The actual UTC times of the observations at a given site do not necessarily correspond to dark times.

from Master level to post-docs and was focused on the simulation and analysis of data with `ctools`, plus a small tutorial also on `gammapy`. The script that I used to generate the XML models was really useful during the school to show how models can be created easily for any source type.

Finally, the files produced for the DC-1 were then provided to the simulation group, which integrated those models in a pointing pattern for the simulation.

3.3 The dataset

In the DC-1 the collaboration has produced a set of simulated high-level science data (so-called Data Level 3 or DL3, in which the reconstructed direction, energy and time are specified for each gamma-ray like shower, that are both real gammas and misclassified charged particles).

The data represent one possible incarnation of the first three years of KSPs, with 1980 hours scheduled for the Southern Array and 1815 for the Northern array, under the assumption of a total of 1100 hours/year/site of observation time. Those hours were distributed among four different KSPs, namely the Galactic plane survey (GPS), the Galactic center survey (GC), the extragalactic survey (EGal) and the AGN monitoring (AGN). The simulations were generated using a simplified scheme for the scheduling of the observations, resulting in the observation schedule in table 3.1. Only the zenith angles of 20° and 40° were simulated (the IRF closest to the zenith angle of the observation was picked) and the energy range between 30 GeV and 160 TeV was used for the simulations (this limit is lower only for AGN, which is 50 TeV).

The outcome of the DC-1 were both the simulations, coming in one dataset per each KSP (with a total size of around 20 GB) and the model of the sky which was used to perform the simulations, in the form of the XML files, file functions for spectra, lightcurves and 3d maps for extended sources: this was done in order to compare the output of the analysis tools with the input models. In the future data challenges the models will most probably not be given, so that the source search will be blind and the data challenge more realistic.

3.4 The close-out document

The data analysis of the simulated data was carried for more than one year and many people were involved in the analysis of the data, reaching one of the main goals on the DC-1. A complete description of the goals, physical models, analysis done, results obtained and feedback from the DC-1 has been reported in a close-out document, which was filled by the different working groups. I'll list here a summary of the close-out document from the "Executive summary" inside it, together with a focus on the analysis that has been done by the members of the Consortium on the extragalactic data.

Catalogue tools: the DC-1 allowed people to build and test pipelines aimed at building source catalogues. When real data will be available, the first task will be the search of gamma-ray emission above an isotropic background of proton-induced showers, together with iterative procedures to disentangle single sources from a diffuse gamma-ray background, which is especially important in the case of the Galactic plane survey in which there's a higher source confusion with respect to higher galactic latitudes. Many procedures have been developed for the need of performing an unbiased source search and the creation of catalogues. Roughly 250 sources were detected, with 80% of them correctly identified (the other 20% being unidentified due to source confusion), which seems in good agreement with respect to the expected sensitivity. Those kind of studies will for sure benefit from other IRFs optimized for angular resolution, since the IRFs produced for the DC-1 are optimized only to obtain the lowest possible sensitivity (nevertheless keeping the angular and energy resolution below the requirements).

GPS specific source search: there are many different types of sources that are expected to be seen by CTA during the GPS, such as hard-spectrum sources (Pevatrons), looking for cut-offs at VHE, pulsars, searching for a pulsed emission at VHE, and periodic emission from binary systems. These models were provided in the DC-1 and the search was done both via blind or targeted methods. Even though these sources have already been detected by current IACTs, no significant detection has been achieved so far: either the models used to generate the sources were inadequate or the observation strategy did not match the periodicity of the sources. These models will for sure be updated in the next data challenges together with better algorithms for the source detection.

Diffuse emission: an important model included was for sure the interstellar emission model, which is particularly important in the case of source search in the Galactic plane survey. The model which was produced was built taking into account existing observations and state-of-the-art cosmic-ray propagation models. An additional diffuse emission component is accounted for by the PWNe and the SNRs that are not clearly disentangled from the background. The contribution of unresolved sources

to the diffuse emission can be as high as 90-95% at TeV energies with an average of true interstellar fraction at 70-80% of the diffuse emission.

Dark Matter: the model produced for the DC-1 had both a line emission model at the Galactic center and a broader W^+W^- signal were searched for. Both a standard ON/OFF analysis and a morphological analysis were performed in a $8^\circ \times 8^\circ$ region around the Galactic center. No signal was detected but the procedure to provide exclusion limits on the annihilation cross section was established. The ON/OFF method failed to perform as well as the likelihood method, with the latter method which proved to be robust enough to constrain the DM annihilation in the $b\bar{b}$ -channel to a level close to the thermal cross-section.

AGN science: two main KSPs fall in this topic, namely the extragalactic survey and the AGN KSP, in which different long-term monitoring of some sources, catching AGN flares and building high-quality spectra are listed as sub-programs. The analysis of the DC-1 data showed that the search procedure for the extragalactic survey resulted in a sensitivity of 1% of the Crab flux for a 500 hours survey, in line with the expected sensitivity (the only bias in the DC-1 was the non-blind search, which was performed at the known position of the sources taken from the DC-1 model). The AGN monitoring task found 16 sources detected in at least one weekly bin among the 24 simulated ones, with 10 sources showing at least 8 weekly detections not accountable to random fluctuations. Four sources (one is shown in figure 3.3), with high redshift (optical depth $\tau > 1$), were used to perform some EBL studies, getting a normalization factor for the EBL in agreement with 1 (the real EBL model), so that there's no bias and the EBL parameters can be constrained from the observation of sources at high redshift.

Tools: ctools and gammapy were the tools used for the analyses of the DC-1 data. The goal of the DC-1 was to test those tools, in order to assess the capabilities and find missing features or possible bugs, testing the output of the analysis with the Monte Carlo truth. Two main analyses now exist in both packages, both the classical background determination (reflected or ring background) one that has been used for several decades from IACT, and the likelihood one, similar to the one used by the Fermi satellite. The general outcome from the DC-1 for what concerns the analysis tools showed that the tools are in a very advanced state and they both can be used to perform accurately high-level analyses of IACT data.

3.5 Conclusion

The work done for the DC-1 has requested the setup of several Python pipelines in order to be able to correctly handle the catalogues from which the population of AGNs was built. The sources have been selected combining data from several catalogues and their

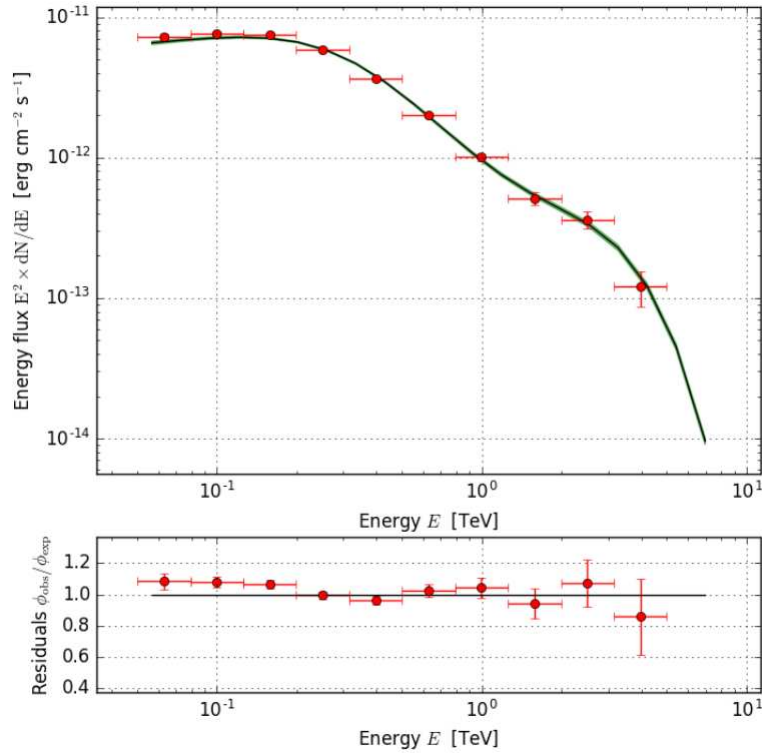


Figure 3.3: spectrum and residual obtained in the full energy range for the source 1FHL J2150.2-1413, detected at $\tau > 1$.

models have been built in order to be simulated according to the simulations schedule and the implemented. After implementing the models, once the simulations became public and they were analysed by the some members of the Consortium, I spent some time cross-checking the results obtained from the simulations against the input models provided by myself.

The work has been developed in the Extragalactic Working Group of the Consortium and it has served as a first benchmark to build the future population of AGNs to be used for the Second Data Challenge which will be followed by the Observatory itself according to the results that must be achieved for this DC-2.

Chapter 4

Prospects for GRB and GW follow-up with CTA

In this chapter I'll present all the work done for the estimation of the detection of GRBs with CTA from the first phenomenological approach to the larger work done for the GRB Consortium Paper. I'll present a pipeline developed for this task and its implementation for the Gravitational Waves follow-up Consortium paper.

4.1 GRB with CTA

4.1.1 Fermi extrapolation

The work on the estimation of the performances of CTA toward the detection of GRBs started with a work based on the extrapolation of the spectra of those GRBs observed by Fermi-LAT having a late time emission.

Sources selection I took the spectra of 10 GRBs (9 reported in [53] with the addition of GRB 130427A), whose spectra are described by power laws, and extrapolated them from the Fermi-LAT energy range to the CTA energy range.

Since GRBs are cosmological sources, in order to do this extrapolation properly, we took into account the effect of the EBL at the highest energies using the model by Franceschini [76]. Even though the power law spectra are expected to vary both in flux and in slope over time, only the flux was varied over time in this first work, keeping the slope constant: the spectral and temporal indexes to be used for the time evolution of the GRBs were taken from [53] and [79]. As a first test, the models of the GRBs were extrapolated at different redshifts, varying only the shape of the spectra according to the corresponding absorption due to the EBL, without varying the flux according to the luminosity distance of the source (see left panel in figure 4.2).

The simulations were done with `ctools`, a software package developed for the high level scientific analysis of the CTA data. Using the IRFs created from the Monte Carlo production (`prod2` for the results shown here), the events that are simulated are the result

of the superposition of the source events and of the background events. The spatial and spectral components of the source and the background models are used to create the list of events in the field of view of the simulation. After the simulation is created, the list of events with their arrival times, sky coordinates and energies are used to fit a model, where both the model of the source and the background are fitted at the same time.

Even though the sources for these simulations have a complex spectra, the model used to fit the simulated data in this first simulations is a simple power law and the output of this fitting procedure is the Test Statistics (TS) parameter.

Test Statistics The Test Statistics parameter is used in ctools to get a measure of the significance of the detection of a source by fitting the data in two different scenarios, with and without including the source in a model. The parameter is defined as follows:

$$TS = 2 \ln \left(\frac{\mathcal{L}^1(\vec{\beta}_j)}{\mathcal{L}^0(\vec{\alpha}_i)} \right), \quad (4.1)$$

where \mathcal{L}^0 is the maximum likelihood for the model without the source at a certain position and \mathcal{L}^1 is the maximum likelihood for a model with the source. TS is a measure of how well the set of parameters $\vec{\beta}_j$ stands in agreement with the data. If $\vec{\alpha}_i$ are very different from $\vec{\beta}_j$, that $\mathcal{L}^1(\vec{\beta}_j)/\mathcal{L}^0(\vec{\alpha}_i) \gg 1$, the TS value is high: a poor agreement between the two sets of parameters means that the addition of the source to the model without it is significant. The higher is TS, the higher the likelihood that a source, described by some parameters, is really present in a certain position. If the two likelihoods, with and without the model, were very similar, their ratio would be close to one and TS close to zero, meaning that the addition of the source to the model has no impact and the data are compatible with the background model.

If we want to understand how significant the TS parameter is, we have to consider Wilks' theorem which states that if we have a large number n of events (with n approaching ∞), if the null model is true, the probability density function of the test statistics parameter TS of eq. (4.1) is a χ^2_{j-i} density for $j-i$ degrees of freedom (with $j > i$). In this calculation j is the number of free parameters in the model with the source and i is the number of parameters of the model without the source (the two hypotheses are nested).

From this probability density function, if TS_{obs} is the value for the TS parameter from two particular models and a certain data set, one can calculate its associated **p-value**, which is the integral of this function from TS_{obs} to infinite: this means that, as χ^2_k goes like $\propto x^{k/2-1} \exp^{-x/2}$, with x the number of events and $k = j - i$, its integral from TS_{obs} to ∞ goes to zero rapidly as ts_{obs} increases.

This means that the higher the value of the TS parameter is, the lower is the probability that the source that has been introduced in the model corresponds to a fluctuation from the background.

The significance of the detection is equal to the square root of the TS parameter only for a model which is described by one degree of freedom, and when the TS parameter is

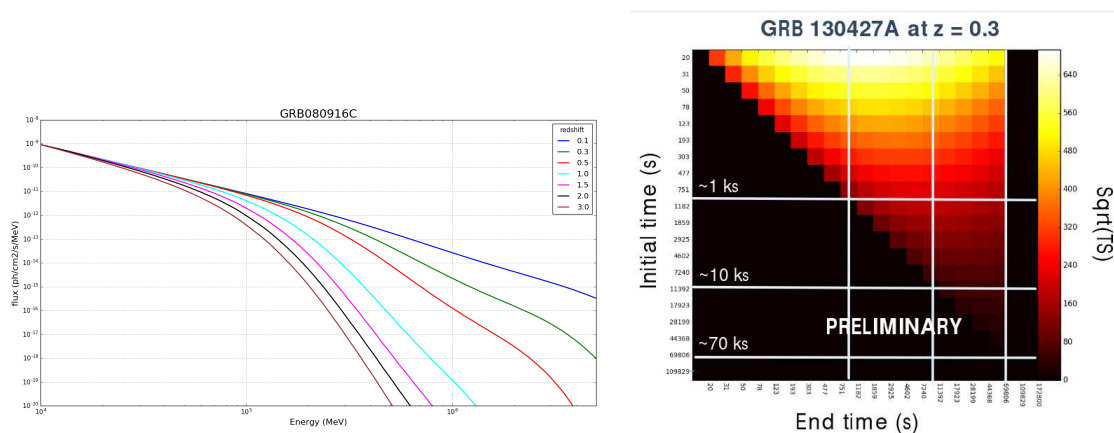


Figure 4.2: *Left Panel:* effect of the EBL on the spectrum of a source when the redshift of the source is changed from 0.1 to 3.0 using the model by Franceschini et al. [76]. Only the shape of the spectra changes according to the EBL, while the flux is not scaled according to the luminosity distance. *Right panel:* the significance of the detection as a function both of the initial time and the end time of the observation. It's very important to follow the GRB as soon as possible, but even if the source becomes visible after several hours, it can still be detected with CTA. The value of the square root of the Test Statistics (TS) parameter, approximately equal to the standard deviation of the detection, is reported in this plot. Figure from [80].

greater than 25, the standard deviation for the detection is greater than 5 and the source is considered as detected. If the number of degrees of freedom is greater than one, the significance of the detection can only be approximated with the square root of the TS parameter: the higher the number of degrees of freedom, the higher will be the divergence between these two values. The higher the significance and the smaller is the difference between the square root of TS and the results that we can from the p-value.

Results These GRBs were simulated several times, using different IRFs, using a simulation window of 2 hours and starting the simulations at different increasing times from the onset of the GRB. Some preliminary results based on these 10 GRBs was reported in [80], where we showed that the significance for the detection of a GRB is higher if the GRB is observed from early times (right panel in figure 4.2). The significance increases with increasing length of the observation reaching a maximum when the source becomes too faint with respect to the background. Continuing the observation after the maximum of the significance is reached is not useful for the detection since more background photons are being included instead of source photons and the signal from the actual source is “diluted” on the higher background.

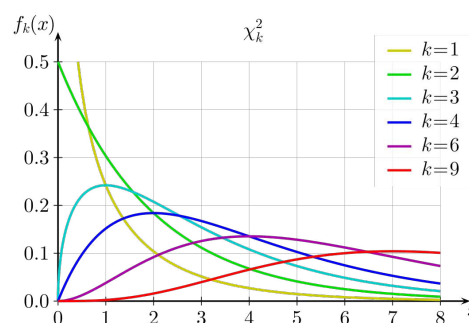


Figure 4.1: χ_k^2 distribution density function for different d.o.f.

The first estimations that we did on the detection of GRBs with CTA were done

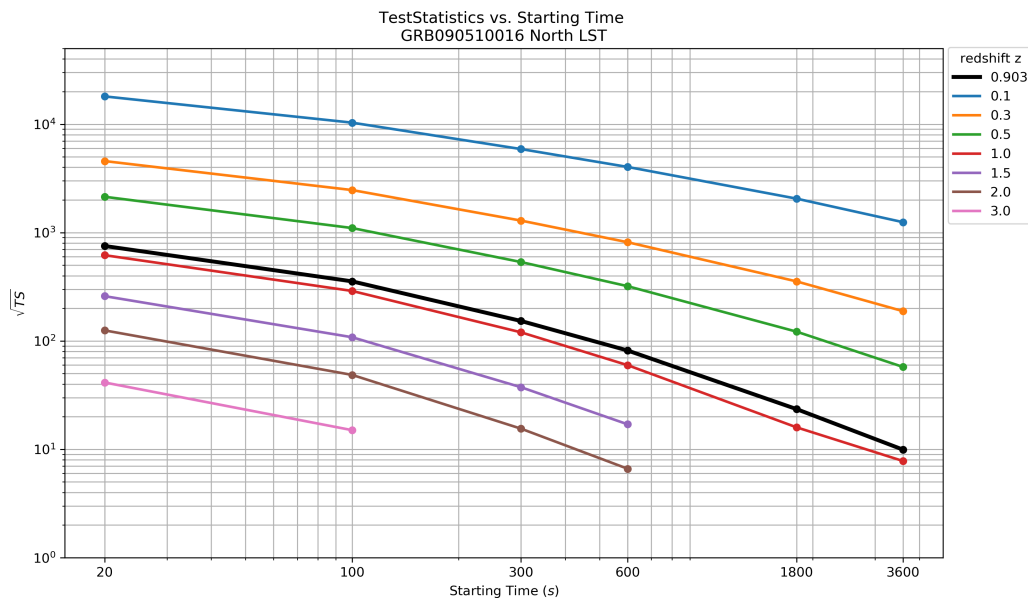


Figure 4.3: The points reported here represents the square root of the TS parameter as a function of the beginning of the simulation, lasting 2 hours. The GRB has been placed at various redshifts (including its real one). The IRF used for the simulations of this GRB is the one of the 4 LSTs in the North site and shows how important it is to repoint to the GRB as soon as possible, especially if the GRB has a high redshift.

with some simplifications concerning the types of IRFs used and the models of the sources placed at different redshifts.

The first IRFs used were coming from the **prod2** and only for the full North and South arrays, while in the following simulations we used new IRFs from the **prod3** for different subarrays of telescopes. I simulated the GRBs both for the North and the South site, for the MST subarray and the LST subarray, starting the simulation after the maximum repointing time of each telescope type. The simulations for the LSTs were done starting from 30 seconds after the onset of GRB, while those for the MSTs were starting at 90 seconds [7]. The outcome of this work stressed the importance of the fast repointing time of the LSTs and their importance, over the MSTs, for the detection of GRBs, due to the lower energy threshold that they can achieve.

Moreover, the 10 sources were simulated at different redshifts not only changing the shape of the GRB according to the different EBL absorption at different redshift, but also considering the difference in the flux due to the distance. For each GRB, its distance was considered as a reference and the square of the ratio between the luminosity distance of the real GRB and the luminosity distance corresponding to the new redshift was used to scale the flux. The significance of the detection drops as expected if the GRB is placed at a higher redshift, both due to a lower flux and to a softer spectrum.

4.1.2 GRB Consortium paper

After these preliminary works based on extrapolations from the GRBs observed by Fermi-LAT, we started working on the CTA Consortium paper whose goal is to determine de-

tection rate for long GRBs that will be observed with CTA and to understand the characteristics of such observable GRBs.

While the model used for the previous works where only phenomenological extrapolations of the spectra observed by Fermi-LAT, the input models to be used for the Consortium paper are computed from theoretical models. Both the modelling of the prompt phase and the afterglow phase at VHE are included in the GRB models.

The synthetic population of GRBs which is going to be produced is such that applying constraints to this population, coming from experiments working at different wavelengths, the detection rate and the parameters of the observed GRBs are reproducing those reported from the analysis of real data. This is an important test because it ensures that the population which is going to be used for the determination of the detection rate of GRBs observed by CTA will be as close as possible to the real population of GRBs. More details on the creation of the synthetic population of GRB and the calibration with the data collected from present and past instruments in many wavelengths is reported in [81].

Time variable spectrum modelling A first dataset of 10 models for the sources have been used to test different analysis methods and compare the results obtained with different pipelines: I've been working on a pipeline developed with ctools whereas other colleagues have been working on the gammapy-based pipeline.

These preliminary models allowed to do some tests for the full simulation and analysis pipeline to be implemented for the Consortium paper. One of the most important ideas which was tested at this state was the creation of the models to be given to ctools for the simulations. Since there's no model yet available in ctools for simulating a source having a spectra which varies in flux and shape over time, I came up with an implementation using already existing ctools functions.

The group of people producing the models is producing a file for each GRB to simulate. The total time range of the simulation (10000 seconds in these files) is divided into 10 log-spaced time bins and a spectrum of the source is provided per each time bin. The spectra are not simple functions, such as power laws, and the flux (in $\text{ph}/\text{cm}^2/\text{s}/\text{MeV}$) is specified for different values of the energy (50 values): the spectra are provided already absorbed with the EBL, using the model by Dominguez [82].

Moreover, in ctools a source can have a light curve specified using a FITS file with the time and the normalization factor used to scale the whole spectra at that time. This possibility of adding a time-dependent normalization factor was used to create an ad-hod light curve as shown in figure 4.4:

The different models for the source are considered as separate sources in ctools, but all of them have the same spatial component, meaning that they are placed in the same position in the sky, which is the position of the GRB. The spectrum of the GRB is specified for every time bin (0–1.2s or 1.2s–3.5s in figure 4.4) and each spectrum is assigned to a different source in the model. The last piece is the creation of the proper light curve for each time bin in order to have only one of the 10 models spatially overlapping being active

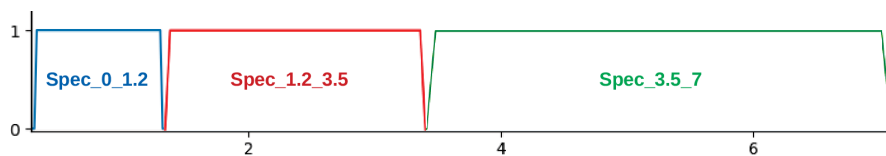


Figure 4.4: time variable spectrum implementation in ctools. Each time interval corresponds to a source which is turned on only inside that particular time interval.

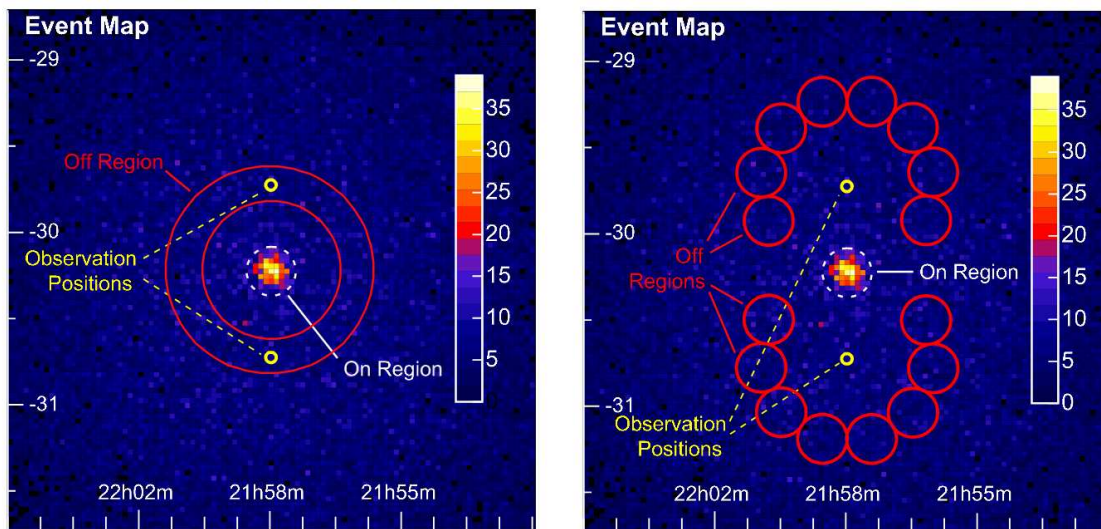


Figure 4.5: ring background model (left) and reflected background model (right). The observations are done in the “wobble mode”, observing the source not on-axis but with a small off-axis angle. Figure from [83].

at the correct time bin. This is done creating a light curve which is 0 everywhere, a part from the time interval in which the corresponding model should be turned on, in which the light curve is equal to 1. A representation of this light curves is shown in figure 4.4 where the first three light curves for the first three consecutive time bins are reported.

ctlike vs on/off The first method implemented in ctools is the 3D likelihood method which uses the model of the source and the TS parameter to compute the significance of the detection of the source over the background: this method is commonly used in the analysis of the data from Fermi LAT.

The second method is a classical technique for IACT spectral analysis and uses the counts in a sky map to compute the significance of a source. The two main methods used to subtract the background are the ring background and the reflected background models, represented in figure 4.5. The idea behind this method is to define ON (source) and OFF (background) regions and then use the counts observed in these regions to compute the significance using the so-called Li & Ma formula in [22] (see example in figure 4.6). In this formula the photon count in the ON region, in the OFF regions and the knowledge of the ratio between the extension of the regions is used to compute the significance of the source detection.

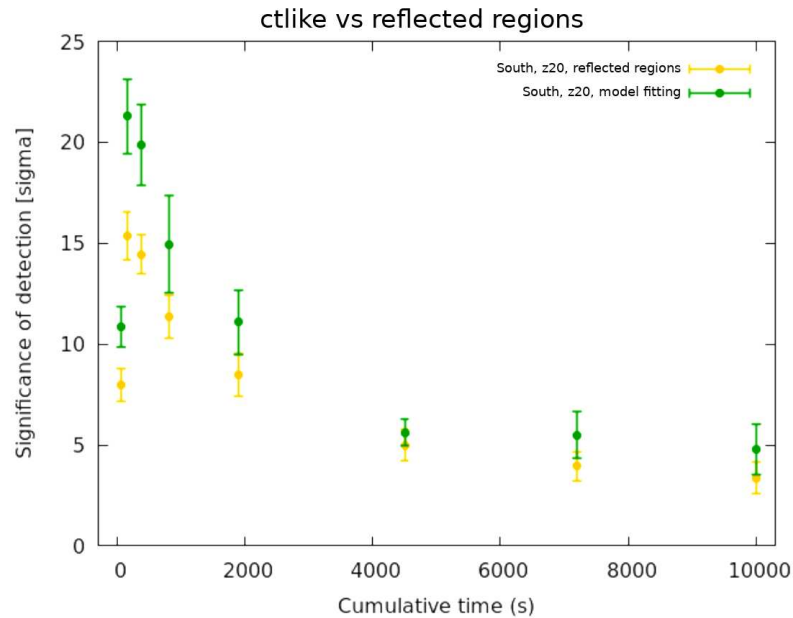


Figure 4.7: comparison of the performances between the reflected background method and the model fitting with clike using a power law with exponential cut off. This result is obtained for one GRB simulated 10 times with different a different seed for the random number generation. The time reported is cumulative from the beginning of the observation.

The size of these regions is particularly important for the ON region since a too small region would not include all the signal and a large region would also include some background. If the pointing directions are too close to the source position, the OFF regions would be contaminated with source signal or the OFF region would not be large enough.

The position of the source must have an offset with respect to the pointing which is used in the simulation, neither too small or too large, with a 0.5° being usually a good compromise: if the offset is too large, the count number would drop due to the lower acceptance of the camera, while a small value would not allow the creation of enough OFF regions.

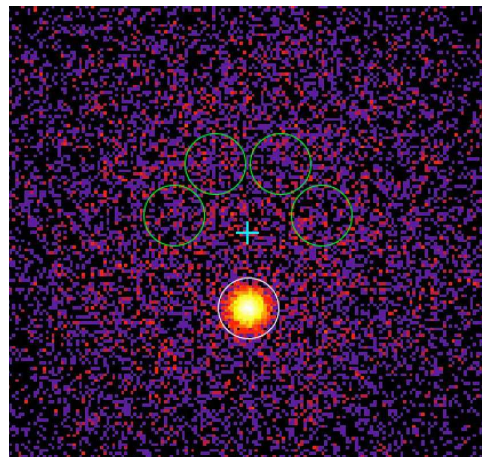


Figure 4.6: sky map from a ctools simulation showing the ON region (in white), the pointing (the cyan cross) and the OFF regions (green circles).

The reflected background method, which is also implemented in gammapy, showed that the 3D likelihood method gives in general a higher significance for the detection (see figure 4.7) since it is taking into account the background and the spectrum of the source and not only the photon count in the sky map.

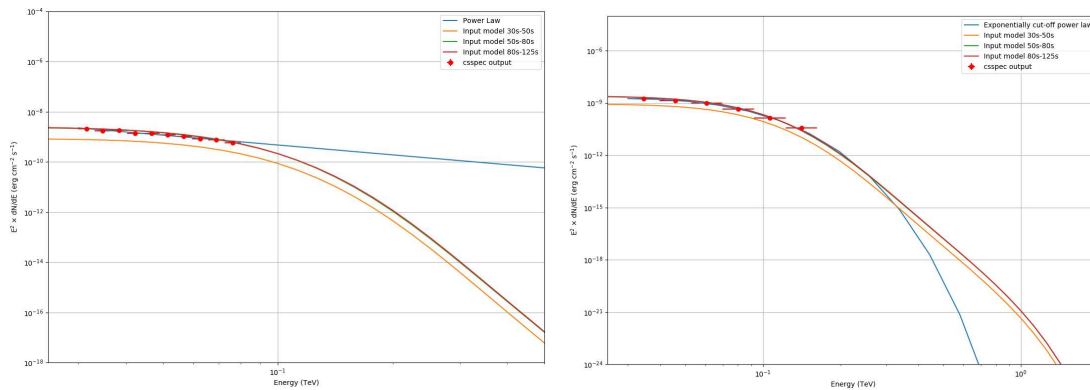


Figure 4.8: result of the fit at low energy (left) and over the whole energy range (right) using the result obtained at low energy. The blue line is the output of the likelihood method, while the orange and red curves are the input models used to describe the GRB in the time interval 30–80s. The red dots are computed with the `csspec` script from `ctools`, which is computing the source spectrum fitting the model separately in each time bin (only low energy on the left and broader energy range on the right).

Fitting other models The models used up to here for fitting the simulated data are power laws, which is a big simplification due to the shape that the spectra have especially due to the effect of the EBL (see left panel in figure 4.2). The power law can be used to fit the data only at the lowest energies because the effect of the EBL is almost negligible, while at the highest energy the effect must be taken into account with a proper model.

Among the models that are available in `ctools`, we decided to fit the simulated data with exponentially cut-off power law (ECPL):

$$M_{\text{spectral}}(E) = k_0 \left(\frac{E}{E_0} \right)^\gamma \exp \left(- \frac{E}{E_{\text{cut}}} \right) \quad (4.2)$$

where k_0 is the prefactor ($\text{ph}/\text{cm}^2/\text{s}/\text{MeV}$), γ is the index of the power law at low energies, E_0 is the pivot energy (in MeV) and E_{cut} is the cut-off energy. In order to find the proper parameters to build this model, we decided to perform two likelihoods, one at low energy using a power law and one on the full energy with ECPL model. The data are first selected in the low energy range, i.e. 30 GeV to 80 GeV, a power law model with default input parameter is fitted on these data and the values for this power law are used to build the ECPL model, using a default value for the cut-off energy (see figure 4.8).

Models update This procedure was tested both on the former 10 models, which was a set of very bright GRB, and on a second dataset with 10 GRBs more representative of the full population of GRBs. These GRBs are randomly taken from the full population, so that not all of them are very bright, and the FITS files for the models are created with a structure decided between the simulation and analysis group and the people working on the synthetic population of GRBs.

4.2 ctools_pipe

The details concerning the modelling of the synthetic population and its calibration with data from instruments observing GRBs, together with a description of the analysis done from the `ctools` and `gammapy` groups is reported in [81].

After all the preliminary tests done until here on many different details of the analysis, I decided to implement a full pipeline based on `ctools` to be used in all the simulations and analysis. The pipeline is fully written in Python (https://github.com/thomasgas/ctools_pipe) and it is used through a main script with proper input flags and the corresponding `yaml` configuration files: this ensures that the code is not touched and only the input files can be changed. Before using the pipeline it is necessary to install other few Python libraries and download also the Python script which is needed to create the `xml` models to be used in the simulations: this script is the same which was used for the creation of the models for the Data Challenge and all the other models that I used in other `ctools` simulation.

The pipeline has two main components, namely the task to work on and the way in which the job which has to be executed is submitted either to a computing centre or locally. There are three different options for the task, namely background simulation, model creation or source simulations and detection. The pipeline was built in order to submit jobs over three different computing centres or locally in any machine, being a laptop or the front end of a computing centre. The general way of using this pipeline is:

```
python ctools_pipe.py --task task_input.yaml --jobs jobs_computing.yaml
```

where the `task_input.yaml` is giving all the necessary input parameters to execute `task`, while `jobs_computing.yaml` is storing all the informations that are needed in order to properly execute the job. Together with inputs in the form of text, numbers or boolean values, the script handles also environmental variables, usually saved in external files.

4.2.1 Jobs submission

The submission of jobs is handled by the `--jobs` flag and the corresponding `yaml` file according to the computing device which is being used:

Local/front-end: the script are executed knowing only the `$MAIN_FOLDER` environmental variable and the path of `ctools_pipe`. These can be stored in an external file and sourced from there.

Trieste INFN Farm: once logged into the computing farm, the user ends up in the front-end and can send the jobs using the `bsub` jobs scheduler. The `yaml` file will be used to select the right queue, the email address to be used to send the output of the job submission and the correct paths for the software, the output and the Python interpreter.

CNAF computing centre: CNAF needs almost the same `yaml` file of the INFN farm in Trieste, a part from a different name of the queue to be used for the job submission.

LAPP computing centre: the MUST computing centre at LAPP uses the `qsub` jobs scheduler. Together with the proper path for the output folder and those of the software that are used, MUST can be used only using the proper environmental variables that must be sourced at every new login on the front-end. Once the script is run, it will print the `qsub` command-line text with all the required flags to the terminal which then can be pasted in the terminal to submit the jobs to MUST: the goal of the script is to create the proper command-line input.

The implementation of the jobs submitting system in the pipeline ensures that any user of the pipeline doesn't have to take care of the details of the submission of a job to a computing centre.

4.2.2 Folder structure

The analysis happens in the `$MAIN_FOLDER` directory and later on develops according to this folder's tree:

```

$MAIN_FOLDER
├── back_sim
│   └── North_z20_0.5h
├── input
│   ├── Event0.fits
│   ├── ...
│   └── Event9.fits
├── input.tar.gz
├── models
│   ├── Event7
│   │   ├── lightcrv .5 lc_000_tin-1.000_tend-1.188.fits
│   │   ├── ...
│   │   └── lc_069_tin-145452.672_tend-172800.000.fits
│   ├── model_Event7.txt
│   ├── model_Event7.xml
│   └── spectra
│       ├── spec_000_tin-1.000_tend-1.188.txt
│       ├── ...
│       └── spec_069_tin-145452.672_tend-172800.000.txt
└── output
    ├── Event7
    └── GRB-Event7_seed10.txt

```

4.2.3 Tasks

Background simulation The list of events simulated in `ctools` from an input model and with a certain IRF, is just one of the possible realization of the simulation which is controlled by a random number generator. A seed value can be specified at the beginning of the simulation in order to get a list of events which is different from the one obtained before with a different seed parameter. This parameter is usually fixed internally in the simulation, so that every time a simulation is re-run with the same input parameters, the same list of events is produced. The simulation is repeated several times, each time changing the seed of the random number generator in order to get a new realization. Each

background simulation will be attached to a simulation of the source and the detection algorithm will be run on this simulated data to get the significance of the detection. The final result will be an average for each result obtained using each time a different realization of the background and of the source.

In order to be sure that a certain GRB can be observed with CTA, the simulations will be repeated 100 times for each GRB, each time changing the value of the seed parameter. Since each repetition of the simulations can take a lot of computing time, we have decided to simulate the 100 realizations of the background just once, leaving them saved on disk, and later on after the source has also been simulated, the event list of the background is attached to the one of the source. The 100 realizations of the background can be reused for all the GRBs with no need of simulating them again for every realization of every GRB. Since the realization of the background depends on the IRF, there will be a different set of simulations saved on disk for each of the IRFs used. The background is created in this way:

```
python ctools_pipe.py --background background.yaml --jobs jobs.yaml
```

and the parameters specified in the `background.yaml` input file are:

```
IRF:
  prod:
    number: 3b
    version: 2          # 0, 1 or 2
    zenith: 20
    site: North
    time: '0.5h'       # 0.5h is valid only for prod3-v2. otherwise use 30m
    pointing: average
    subarray: LST      # only for prod3b: LST/MST/SST/MSTSST/FULL/TS
    TS: 0              # only for prod3b: 1 (yes), 0 (no)
sim:
  realizations: 5      # will be launched as 10 separate jobs
  radius: 3           # in degrees
  time:
    t_min: 0 s
    t_max: 1800 s
  energy:
    e_min: 30 GeV
    e_max: 5 TeV
```

where the two sections for the choice of the IRF and the actual simulation can be seen. The background is simulated by default around RA, DEC = (0°, 0°). The script handles the IRFs from the different Monte Carlo productions and simulates the background.

Model creation The models of the sources are created with the same method described previously with the difference that the time bins are not 10 but 70, starting from 1 second after the burst up to two days. The models are created with the simple command:

```
python ctools_pipe.py --models model_input.yaml --jobs jobs.yaml
```

and the parameters specified in the `model_input.yaml` input file are:

```
models:
  type: GRB          # can be GW or GRB
  scaling: 1
  pos:
    ra: 0           # value in deg. null for None: will be extracted from fits header
    dec: 0.5        # value in deg. null for None: will be extracted from fits header
  input_data:
    path: $MAIN_FOLDER
    version: 0      # keep 0: versions of models not implemented yet
  output: null      # $MAIN_FOLDER/models. null for None. Default is "models"
  max_models: 10   # type null if all models have to be processed
overwrite: yes     # yes --> re-create models. no --> keep already created models
```

where the source type to be simulated can be seen (either GRB or GW). The position of the source can be selected together with other parameters for handling the creation of the model, which has been only implemented to be executed on a laptop or the front-end of a computing centre, without submitting any job. The goal of this step is to create the input file for the script used to create the `xml` models.

Source simulation and analysis This is the biggest part of the script since it is responsible of loading all the inputs, doing the simulation of the source splitting this simulation in the various time bins in which the analysis is performed and finally the detection of the source is computed using different methods. The simulation is done with the simple command:

```
python ctools_pipe.py --simulation simulation.yaml --jobs jobs.yaml
```

There are three methods implemented in the `ctools_pipe` that are used to estimate the significance of the detection, all of them derived from `ctools`.

Standard Likelihood: this is the method described above which involves a two-step process. In the first step the low energy component of the source is fitted with a power law and using the likelihood method and the value for the output model obtained in this step are used to build the model for the fitting procedure happening in the following step.

ON/OFF Likelihood: this method combines the classical ON/OFF method with a spectral analysis creating the ON regions and OFF regions but not stacking the energy informations, so that the counts can be determined in each energy bin. As for the previous method, also this method uses the two-step process described before to build the model from a fit of the low energy data.

ON/OFF method with ring background: this is the classical ON/OFF with the ring background subtraction method. For each pixel in the map the ring method estimates the background from a ring centered at the pixel. This method doesn't need any model of the source because it is based only on the photon counts in the sky map.

All the parameters that are needed to use these detection methods can be tuned using the external configuration file.

```

IRF:
...
source:
  type: GRB                # can be GW or GRB
  pointings_path: $MAIN_FOLDER/pointings # only for GW
  max_sources: 1
ctobssim:
  radius: 5                # in degrees.
  time:
    t_min: 30 s
    t_max: 120 s
  energy:
    e_min: 30 GeV
    e_max: 10 TeV
  realizations: 1          # per each model
  background_path: $MAIN_FOLDER/back_sim
  models_in:
    xml_path: $MAIN_FOLDER/models
    fits_path: $MAIN_FOLDER/input
ctselect:
  time_cut:
    mode: 'log'           # this can be log or lin
    obs_mode: iter       # "cumul": t[0] --> t[i]. "iter": t[i] --> t[i+1], "all": t[i] --> t[j]
    t_slices: 2          # 0 --> keep all time interval...t_min and t_max are ignored
    t_min: 30 s          # write N/A to use the same as the simulation
    t_max: 120 s         # write N/A to use the same as the simulation
ctlike:
  counts: yes            # simple Li&Ma (uses ctskymap RING method). to disable --> no.
  pars_counts:
    ra_center: 0
    scale: 0.02          # deg/pixel scale of the image
    roiradius: 0.2
    inradius: 0.6
    outradius: 0.8
    iterations: 3
    threshold: 5
  ctlike-onoff: yes     # onoff with ctools. Needs to create a input model. to disable --> no.
  ctlike-std: yes       # standard ctlike. Needs to create a input model. to disable --> no.
output: $MAIN_FOLDER/output

```

The IRF selection is the same of the one used in the simulation of the background. Under `ctobssim` there are the parameters for simulation (radius of the simulation, time range and energy range), the number of **realizations** per each GRB, the path of the previously simulated backgrounds and where to find the models of the source to use in the simulations. `time_cut` under `ctselect` is used to specify the time binning to use in the simulation. “log” or “lin” in `mode` are used to create linearly or logarithmically spaced time bins.

with increasing time after the time of the merger.

The first pointing starts from three minutes after the GW events has reached the interferometers, which is the average latency that the online pipelines of the LIGO-Virgo Collaborations need to sent the alert. The sky map from the BNS catalogue together with the durations of the pointings are combined to create the pointing pattern, a list of pointings which will be used to observe the sky map in a way in which the probability of detecting the source is maximised. Since the flux of the source is expected to be rapidly fading, the sooner the source is pointed, the higher are the chances of a detection. The maximization of the coverage of the probability sky map is taking into account not only the 2D GW localization maps but also the 3D distribution of galaxies which could plausibly host such events [84].

The last part of this work (the green box in figure 4.9), the one that I've been working on, combines the physical model of the source with the pointing pattern and the durations of the pointings to estimate the performances of CTA on the detection of the emission coming from GW events. While the GRBs are simulated on-axis, meaning that they are observed at the centre of the camera, in the GW follow-up the source can happen in any position of the field of view, meaning that the source has to be search in a blind way across the sky map produced in each pointing.

Another difference with respect to the analyses done for the GRB paper is the distance of the BNS mergers that are expected to be observed with the interferometers, as reported in figure 2.17. Since these events are expected to happen at a smaller distance with respect to the long GRB, the absorption of the highest energy photons due to the EBL is not expected to deeply affect the results. While the LSTs are very important for the observation of long GRBs, it will be important to do the follow-up of GW events using the full array, taking into account that the different telescopes types have different repositioning times.

4.3.1 Pipeline for GW follow-up

The pipeline “`ctools_pipe`” developed for the GRB Consortium paper will be used also for the GW follow-up paper with some modifications especially for what concerns the simulation of the source and the detection methods to use. This results from this pipeline will be compared with the results obtained in [84], where the GW follow-up pipeline has been implemented using `gammapy`.

The simulation of the background will work as in the case of the GRB paper and the creation of the models of the short GRBs will also be similar. The strategy for the simulation of the source and its blind detection is sketched in figure 4.10. This scheme is not intended to be an implementation of a real-time analysis but more an illustration of the workflow to use in `ctools_pipe` to estimate the detection rate of these phenomena and the related parameters.

The first step in the analysis is loading the FITS file of the BNS (used basically only for its RA-DEC position) to simulate, the pointing patterns and the duration of each pointing and the main loop over all the pointings begins. The source is expected to be

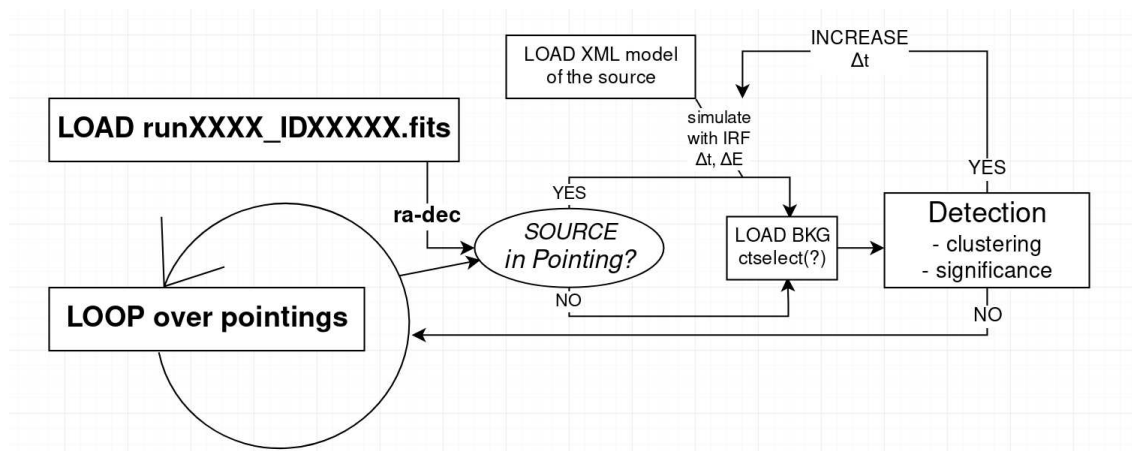


Figure 4.10: a schematic representation of the part of pipeline, to be used in the GW follow-up paper, which is taking into account the simulation of the source and the estimate of the detection. The pointings are considered as separated observations and they are not merged.

at least in one of the pointings, or in more than one if this is happening at the interface between two pointings. If the source is in the field of view of the telescopes, then the script loads the `xml` file of the model (the sGRB model) which is needed by `ctools` and using the proper energy range and IRF according to the height of the source above the horizon, the source is simulated for the duration of the pointing: a realization of the background for the same energy range, time interval and IRF is attached to the simulated source. The sum of the source and the background is passed to the detection algorithm, which can use different methods to do a blind search of the source in the field of view. A positive detection means that the source has been observed and that the observation will continue, simulating the source for a longer time interval and attaching more background, while a negative detection, even though the source is in the field of view, means for example that the source is too weak to be detected.

The field of view which has been used for the simulations so far is not the full field of view of the telescope (4.5° for the LST) but the nominal field of view, corresponding to half of this value has been chosen. The radius of the pointings has been determined using this value and the consecutive pointings have not been merged. The pipeline will be used in the same way for regions where the source is expected to be and where no signal is expected and the algorithm will be run in the same way. If a background fluctuation is picked as a hot spot due to some fluctuations, continuing the observation with the same pointing direction should cancel out this hot spot.

4.4 Conclusions

The expertise accumulated over the last years on the development of data analysis pipelines in Python and the knowledge of the high-level analysis tools have been important for building the final pipeline to be used for the two GRB and GW Consortium papers.

Since the amount of work to be done for these publications is very large, the work is still in progress. The results coming from the pipeline implemented in `ctools` will be cross-checked with those obtained with a similar pipeline implemented in `gammapy`. Some theoretical input models are still missing but the pipeline is almost ready to process the simulation for the GRB paper and the GW paper.

Chapter 5

CTA reconstruction pipeline

The main software packages that have been used so far for the creation of CTA IRFs, will not be used again for the next Monte Carlo production or for the analysis of the CTA data. One possible replacement for these packages is **ctapipe**, a Python framework for the low-level analysis of the CTA data. In the first section I'll go through the implementation of the **ctapipe** framework, highlighting the major components and methods that can be used to do a simple pipeline for the data analysis of a Monte Carlo simulation. I'll also go through the contribution that I made to the code. The second section is focused on the reconstruction methods that are implemented in **ctapipe**, from the more classical ones to the more advanced ones like the template-based method which uses a library of camera images from Monte Carlo simulation to reconstruct the parameters of the incoming particle. The method, called ImPACT, has been implemented in **ctapipe** some time ago and I used it to do some analysis on the array of telescopes. I then adapted the scripts to use the same analysis for the event reconstruction for the prototype LST-1, the first LST which is presently in the commissioning phase in La Palma. I'll present the previous works on this topic, in which a semi-analytical model for the particle shower was used, and in the other section I'll describe the **ctapipe** implementation of ImPACT. I'll then explain the idea behind Machine Learning and Deep Learning techniques applied to the event reconstruction.

In the last section I will present the work done for the creation of a 3D displayer to be used in **ctapipe** to create a rendering of the telescopes together with the images on the camera, the reconstructed shower parameters and the various frames used.

5.1 **ctapipe**

ctapipe is a framework which is being developed for the processing of the low-level data coming from CTA [86], either simulated data for the creation of the IRFs to be used with the science tools, or for the analysis of real data, such as those being collected by the first prototype of the Large Size Telescope, the LST-1, in commissioning phase in La Palma.

The **ctapipe** package was created as a core library to be used later on in a full pipeline

to process the CTA data. It contains both core data structures and algorithms for building such a pipeline and its development is driven both by lessons learned from other experiments and from the basic requirements of CTA.

The development of this library is done via *GitHub*, a very powerful open-source versioning system used world-wide mainly in software development for collaborative projects: this system allows people to work on some features for a library, commit them explaining what has been done, create a “Pull Request” with the files they have been modifying and then merging their work, if accepted by other developers (at least two members of the core developer team in **ctapipe**), with the “master branch” of the library. This versioning system is also used to have a discussion in a Pull Request, it can be used to raise issues regarding bugs or discussions for implementations of missing features. Together with the main code, a documentation page is created with GitHub, where tutorials and examples are placed together with the list of authors, a change log between the various releases. Different tools are installed in the GitHub project of **ctapipe** including tools for continuous integration, unit-test coverage, code quality checking and automatic creation of the documentation.

The **ctapipe** framework can be installed using **Anaconda** (or **miniconda**), which allows to create different Python environments for different projects, specifying the right version of each package in each environment, which is particularly easy for testing software between multiple platforms making sure that all of them are using the same libraries at the same versions.

5.1.1 Framework implementation

The programming language chosen for the development of such framework is *Python*, due to its adoption by a large part of the scientific community, its easiness to use, the large library of already existing code ranging from data access to astronomical and statistical calculation. Packages on which **ctapipe** heavily depends from are very common and have become a standard on science and astronomy (see figure 5.1). *NumPy* [87] is very useful for the **numpy.ndarray** data structure with all the related methods. It’s very easy to perform complex computation even with multidimensional arrays while keeping the code fast to implement, easy to read and to debug. The *SciPy* [88] library comes with many useful methods for data analysis, such as integration, interpolation and a statistics module. *Astropy* [89] is a really powerful library which is useful not only for the astronomy community, but by the scientific community in general, due to its modules for data access, coordinates usage and conversion between different units. *Matplotlib* [90] is the main Python plotting library for plotting and visualizing data, making it easy to create simple plots and allowing for complex representations. *pandas* [91] is another widely used library for reading and manipulating of many data structures, with a great number of tools to easily manipulate them. *Scikit-Learn* [92] is a library built on top of NumPy, SciPy and matplotlib for data mining and data analysis coming with many Machine Learning

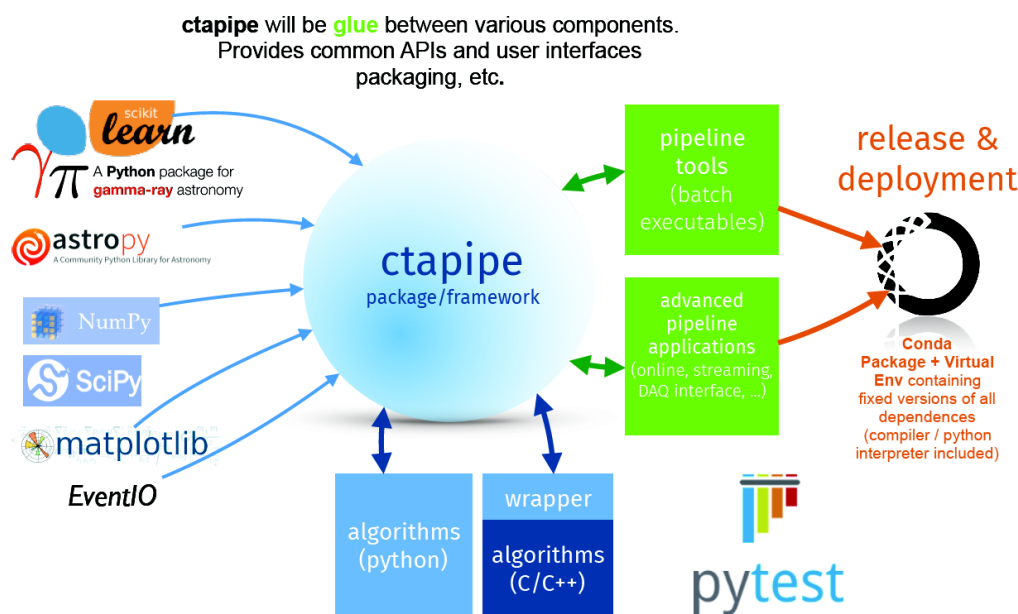


Figure 5.1: sketch of the **ctapipe** framework (credit K. Kosack).

algorithms, classification, regression, clustering and much more. Another very important library which has been recently developed is *pyeventio*, which is used for reading the data coming from the simulations, which replaces an older library written in C.

The framework is built in a top-bottom approach, with an implementation in Python for all the algorithms which can be later on speed-up and optimized thanks to many tools and libraries that have been developed around Python.

The `numpy.ndarray` is the basic data structure, which is very useful for the data manipulation of arrays of numbers without having to deal with for loops: it is reasonably fast, especially if particular versions of *NumPy* are used, such as those optimized for Intel processors.

A first speed-up to the NumPy implementation of a certain method can be achieved using special Python decorators of the Numba package: these decorators, used just before the function definition, are really useful especially in the case of pieces of code in which for-loops are needed or when complex NumPy arrays and functions are used. Numba translates Python functions to optimized machine code at runtime, approaching speeds achievable by lower-level programming languages, such as Fortran or C, without the need of replacing big portions of the code.

Other optimizations includes the usage of *Cython* [93], which is a programming language that makes writing C extensions for the Python language as easy as writing in Python. The source code is translated into optimized C/C++ code and compiled as a Python extension module. This is used in the *pyeventio* library¹ to maintain a python-like code with the execution speeds similar with respect those achievable with C. Other optimizations to the code includes writing C/C++ code directly, which is then wrapped in order to be used in Python.

¹<https://github.com/cta-observatory/pyeventio>

5.1.2 Analysis steps in `ctapipe`

The methods and classes implemented in `ctapipe` are still going through a continuous and rapid development, and the latest major release with breaking changes from the previous 0.6.2 version, is version 0.7.0 [94]. The implementation of the various methods that I'll present here are related to the last version, released on the 30th of August 2019.

I'll present here below the general steps and their implementation in `ctapipe` for the low-level data processing of CTA, similar to those of other existing IACTs, while more advanced analysis methods, i.e. the template-based one, will be presented in the next section.

The names of the various data levels used in `ctapipe` come from the CTAO hierarchical data model reported in appendix B: the data levels in the low level event reconstruction range from the raw data, the R0 data level, to the DL3 data level, each new level being the result of the processing of the previous data level. This last data level is a list of reconstructed events with all the reconstructed photon parameters that are then used with high level science tools to create sky maps, spectra, light curves, etc.

Data loading

The simulated files from the Monte Carlo production are loaded in `ctapipe` using the `simteleventsource` in the `ctapipe.io` module, which is using the library `pyeventio`. This library has been developed to read the data of the Monte Carlo productions coming from `sim_telarray` and is being updated with new releases of such code. This library came from the need of having a more memory efficient data loading and have the possibility of opening multiple files at the same time, which was not possible with the previous reader based on the `hessio` library.

Together with simulated data, also real data from existing telescope prototypes can be read with `ctapipe`, provided that the team responsible for the telescope itself creates an external `EventSource` plugin for `ctapipe`. An example is the plugin for reading the data coming from the LST-1 prototype ².

The loading of the data is easily done from the `ctapipe.io.event_source` function, which is an helper function from the `EventSource.from_url` of the `EventSource` class:

```
from ctapipe.io import event_source

input_file = "example_gamma_file.simtel.gz"
source = event_source(input_file, max_events=10, allowed_tels={1,2,3})

# loop over the events
for event in source:
    ...
```

which is automatically detecting what type of file we are giving as input (between those that `ctapipe` can read) and it is loading 10 events (as a simple test) from this `input_file`,

²https://github.com/cta-observatory/ctapipe_io_lst

which have triggered one or more of these telescopes. A useful parameter for this functions is `allowed_tels` which is used to select the telescopes from their IDs. This is particularly useful because otherwise all the telescopes of the array present in the simulation are used, but they might not be needed if only a subset is needed for the analysis. For example, in the divergent pointing analyses, I was selecting only the MSTs in the simulations, with IDs from 5 to 19.

All the data are stored in a hierarchical structure of **Containers** that store data items along with their metadata, such as unit, description and default value. It's useful to have such data structure with metadata because each field in the data is associated with its description.

At the end of data loading we have all the raw waveforms (R0 data level) from the cameras of the triggered telescopes for the selected events. In the R0 data level there will be one image (the signals from all the PMTs) per nanosecond (30 in total are present in the prod3b simulations) per each gain channel (high gain and low gain for the LST) (see figure 5.2). The raw data per each telescope will be a matrix with dimension $(N_{\text{time_slices}}, N_{\text{PMTs}}, N_{\text{channels}})$.

Calibration and integration

The data loader is also responsible for calibrating the raw data transforming the signal from the R0 level to the R1 level. The signal in the R0 data level comes from the ADC counter, so that the signal in each pixel, per each gain channel, per each time slice is recorded as an integer number. The pedestal for each pixel is subtracted and the signal in the R0 data level is converted from ADC counts to photoelectrons to get the R1 data level: this is done per each gain channel and per each time slice (see figure 5.3).

The next step is the reduction of the data volume, which is particularly important in a real scenario in which it is required that only 3% of the pixels are kept, on average, in the DL0 data level. This means that there must be some procedure, which is currently being investigated, to have some data reduction, with some lossy method which saves only the pixel with signal and discards those having only noise. Together with the data volume, the gain selection happens from the R1 to the DL0: a gain selector is implemented in `ctapipe` and, if no information is provided, the information from the high gain channel is copied from R1 to DL0 (right image in figure 5.3). It's important to mention that even though `ctapipe` is expected to start the analysis from the DL0 level once real data will be available, in order to produce such data level from the simulations, these steps are still mandatory and must be done in `ctapipe`.

At this point there is one calibrated image per each time slice, which will be further analysed to get the DL1 image. A gain channel is selected per each pixel (default is high gain channel) and the signal in each pixel is then time integrated and the readout counts are converted to photoelectrons so that from the DL0 data level we obtain only one image with the number of photoelectrons per pixel (left image in figure 5.3). There are different

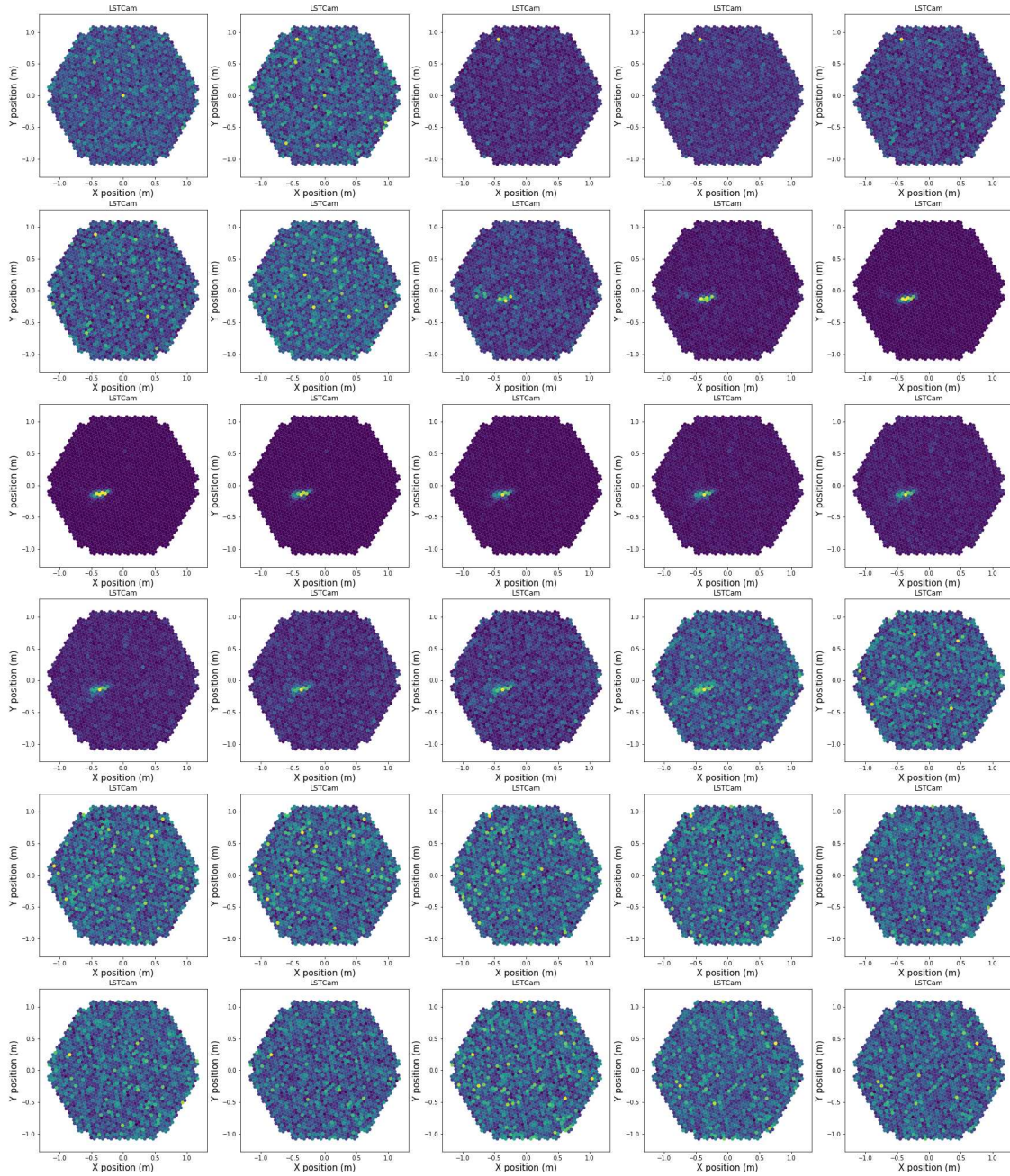


Figure 5.2: the figure shows the R0 waveforms produced for the high gain channel of the LSTCam triggered by a gamma-ray at 250 GeV: there are 30 samples, 1 per ns, per each gain channel.

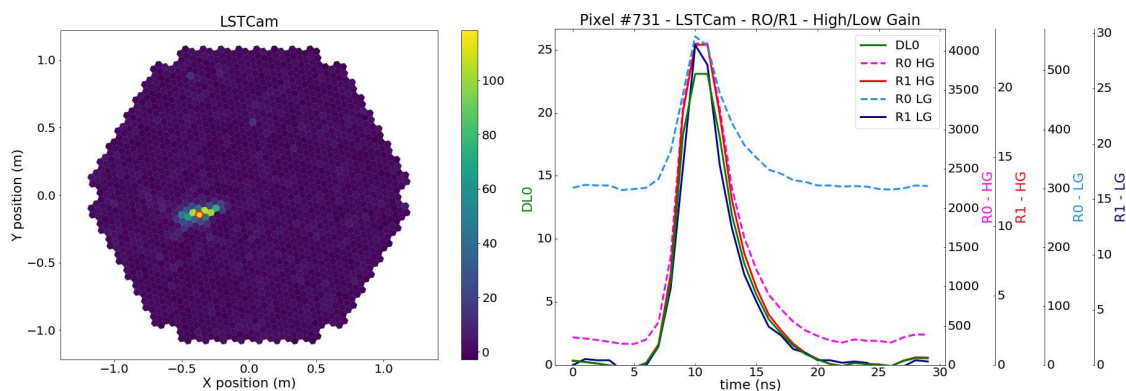


Figure 5.3: the two channels (high gain waveforms for the highlighted red pixel are shown in figure 5.2) are combined in order to have only 1 waveform per pixel, and the charge in each pixel is then integrated over time in order to have just one final image (**DL1** level) with the number of photoelectrons per pixel (left panel). The right panel shows the intensities recorded in the pixel highlighted in red in the left image for different data levels and channels: the right scale is referred to the **DL0** data level where just one channel has been selected, whereas the scales on the left are referred to the high gain channel (**RO** and **R1** data levels) and low gain channels (**RO** and **R1** data levels). This is the brightest pixel in the **DL1** image and is saturated in the high gain channel (which has a maximum value of 4095). The pedestal is around 300 for both channels.

methods implemented in `ctapipe` to do the time integration of the signal in each pixel, the default being the `NeighborPeakWindowSum`. This method looks for the peak of the signal in each pixel on the a 30 ns time window (as it has been simulated in prod3b) and integrates the signal in a small window around this peak. The DL1 data level contains also the peak arrival time in each pixel (the time at which the signal in that pixel reaches the maximum), which can then be used in the some cleaning procedure.

The signal in the DL1 data level is used with the default values in this way:

```
from ctapipe.calib import CameraCalibrator
```

```
# ...event loading...
```

```
calibrator = CameraCalibrator()
calibrator(event)
```

and another image extractor would be an optional parameter in the initialization of the `calibrator` object, instance of the `CameraCalibrator` class.

Image cleaning

The DL1 image is cleaned with a two-threshold tail-cut algorithm, whose parameters are tuned in order to properly select most of the Cherenkov signal in the camera, without including the signal coming from the Night Sky Background. The idea is to include “picture pixels” with a signal greater than a certain “picture threshold”, and to include also “boundary pixels” with a signal greater than a “boundary threshold” (lower than the

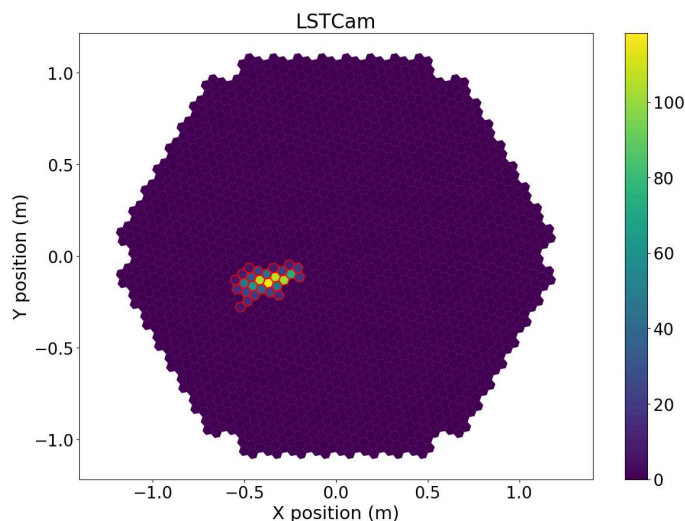


Figure 5.4: pixels passing the tail-cut cleaning highlighted in red. All the other pixels are set equal to zero.

picture threshold) and having at least a neighbour “picture pixel” (see figure 5.4).

The output of the cleaning algorithm is boolean mask with **True** and **False** with the same dimension of the image dimension, which is the number of pixels. The camera informations (extracted from the Instrument Container) containing the pixel position and the neighbour pixel information, the DL1 image and the values of the threshold parameters are used to create the pixel mask:

```
from ctapipe.image.cleaning import tailcuts_clean

# ...data loading and calibration...

tel_id = 4
camera = event.inst.subarray.tel[tel_id].camera
dl1 = event.dl1.tel[tel_id]
clean = tailcuts_clean(camera, dl1.image, boundary_thresh=3,
                      picture_thresh=6, min_number_picture_neighbors=2)
```

The `clean` mask is array of boolean values where the pixels passing the cleaning are marked with **True** and those not passing the cleaning are marked with **False**.

Some cleaning methods can also include the time at which the signal of the pixel is maximum in order to perform a better cleaning. The cleaning used in FACT has been implemented in `ctapipe` (`ctapipe.image.fact_image_cleaning`) and the cleaning of the MAGIC telescopes (from [3]) is also being implemented in `ctapipe`.

5.1.3 Contribution to `ctapipe`

I contributed to the development of the `ctapipe`, becoming a member of the core developers team which is responsible for reviewing the Pull Requests (PRs) on the GitHub

repository of **ctapipe**. Only members of the core developers team can approve them and at least two reviewers are needed to accept a PR and to merge it into the main branch.

The general rule when changing some parts of a piece of code is to add some units tests to make sure that the new piece of code is not breaking other parts of the code and it's working as expected. These tests can be either run locally before committing the code and are also run by GitHub for every new PR in order to check and show to the other developers if the PR is properly written and it can be merged.

For what concerns my contribution to the development of **ctapipe**, apart from smaller PRs, those that have brought some major contribution are related to coordinates, plotting and the algorithms used for the direction reconstruction. The coordinates system was a bit difficult to understand and had some bugs to solve, so together with another colleague we refactored the coordinate system, making use of the functionalities offered by the Astropy package. This work led to the preparation of a Jupyter notebook that has been inserted in the tutorials in **ctapipe** and it is used as a reference for the users.

I've started my contribution to **ctapipe** fixing a bug in the **ArrayDisplay** which was not working for plotting a reconstructed event from a diffuse gamma ray simulation due to usage of the wrong angle for plotting. The hillas ellipse has two peculiar angles, ϕ being the rotation angle with respect to the camera centre and ψ the rotation angle with respect to the x-axis of the camera (see figure 5.5). Since the wrong angle was used for plotting, I fixed this and added the possibility of plotting a line for each telescope and not only an arrow, since the direction is easily obtained from the major axis of the ellipse but the direction is sometimes flipped by 180° , plus correcting for a bug in the plotting of the labels of the telescopes and the range of the arrows.

The most important PR that I did was a refactoring of the three reconstruction methods that are implemented in **ctapipe** (see next section), mainly driven by the necessity of checking that the reconstruction was properly working for the reconstruction in the divergent pointing mode. While working on the unit tests for the **HillasIntersection** class, I found out that some important tests were missing and that there were some bugs in the code: I fixed the bugs in the code and wrote the proper units tests to make sure everything was working as expected. The **HillasReconstructor** was changed for the divergent pointing and also the parameters to use in the function have been changed in order to be able to use it easily both for normal and divergent pointing. The **HillasIntersection** has been changed so that the same inputs can be used for both reconstructor. I also moved a set of warnings, which were implemented only for one class to the common class of the Reconstructors, so that a new reconstruction method that might be implemented can use the same warnings and should be used with the same input parameters. The warnings are raised to the user if the input data are not correct, such as when the number of telescope in the reconstruction is not two or if one image is too small.

This change to the reconstruction classes was important because allowing them to be called with the same input parameters make it easier to compare their performances in a benchmarking procedure. The reconstructors (here the **HillasReconstructor** is used as

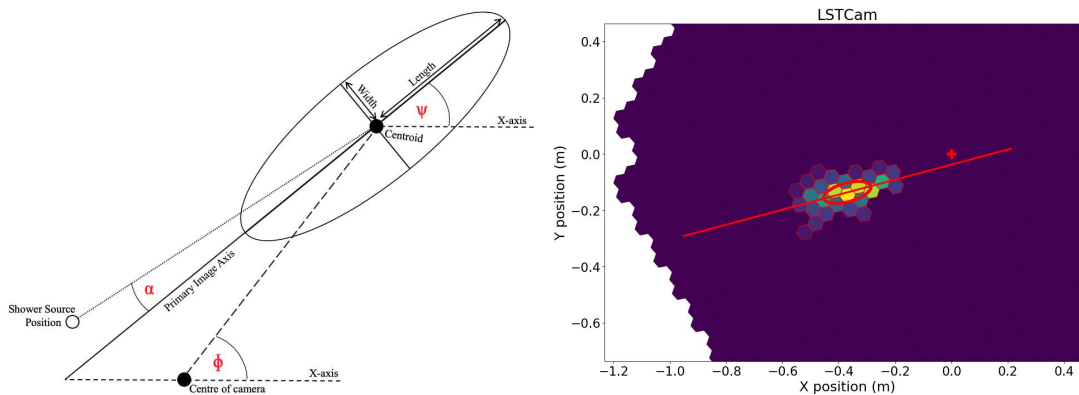


Figure 5.5: *Left figure:* image showing the different parameters that are computed during the Hillas parameterization (credit to J. Watson). *Right figure:* zoom on the pixels passing the cleaning with the ellipse of the Hillas parameterization superimposed and the major axis of the ellipse. Since this is a gamma-ray simulated on-axis, the major axis of the ellipse approximately intersects the center of the camera (the red cross), which it almost does: a fine tuning of the cleaning would improve this result.

example) can be used in this way:

```
from ctapipe.reco import HillasReconstructor
...
reco = HillasReconstructor()
stereo = reco.predict(hillas_containers, event.inst,
                     array_pointing, telescope_pointings)
```

where `hillas_containers` is a dictionary storing the hillas parameters computed per each telescope, `event.inst` contains informations such as the telescopes' positions, the `array_pointing` is the pointing direction of the telescopes and `telescope_pointings` is an additional parameters to use in case of divergent pointing, where each telescope have a different pointing direction. If all the telescopes have the same pointing direction, `telescope_pointings` is not needed and the standard reconstruction method will be used, with no correction due to the different pointing directions of the telescopes.

The ImPACT class was also modified to fix some bugs and to implement some changes needed in order to speed-up the computation (see section 5.2.2).

5.2 Reconstruction algorithms

5.2.1 Hillas parametrization

The resulting image for a gamma ray primary particle after the cleaning is essentially an ellipse, whose moments can be computed using the charge of each pixel as a weight: this step is called Hillas parametrization [2]. The result is a set of parameters, namely the “size” of the ellipse (the sum of the signal deposited in the pixels passing the cleaning), the position of the center, the width, the length, the angle between the major axis and a reference axis in the camera respectively, the skewness of the ellipse and the kurtosis

(see figure 5.5). In order to compute the Hillas parameters only using the pixels that have been selected in the cleaning routine, the `camera` and `d11.image` are passed as “masked” (square brackets in Python) meaning that only the positions and charges of the selected pixels are used:

```
from ctapipe.image import hillas_parameters
...
hillas = hillas_parameters(camera[clean], d11.image[clean])
```

The computation of the leakage parameter (amount of signal deposited in the border pixels of the camera), the timing parameter and the number of islands (which counts the islands signal after the cleaning) functions are also implemented.

Stereo reconstruction

Given the positions of the telescopes on the ground and the hillas parameters of the ellipse in the camera, a stereo reconstruction can be performed for each event in order to find the impact point of the shower on the ground, the incoming direction of the source in the sky and the height of the shower maximum h_{\max} (or x_{\max}).

There are two reconstruction methods available in `ctapipe`, in the `ctapipe.reco` module, namely the `HillasIntersection` class and the `HillasReconstructor` class.

HillasIntersection The `HillasIntersection` class, similarly with respect to what it is implemented in H.E.S.S., does a 2D reconstruction via a line-intersection method in the common frame between all the telescopes, called *NominalFrame*. The major axis and the position of the centre of gravity of the ellipse in each camera is projected into this frame and the reconstruction is done with a weighted average between all the possible pair-wise intersections.

The *NominalFrame* is related with the horizon frame, which is the local reference frame of the observing site so that once the position is reconstructed in the *NominalFrame*, it can be projected into the sky to have its altitude and azimuth. The reconstruction of the impact position in the ground is done in a similar way, using the moments of the ellipses and the telescopes positions on the ground.

The frame used for this reconstruction is the `TiltedGroundFrame`, perpendicular with respect to the pointing of the array: the telescopes’ positions are projected from the `GroundFrame` in which they are defined to this frame, and then the lines passing per each telescope, tilted according to the rotation of the ellipse in the camera, are drawn in this frame, intersected pair-wise and averaged.

The h_{\max} is computed geometrically under the assumption that the shower maximum lies at the brightest point of the camera image: this height is estimated for each telescope taking the ratio between the distance between the center of gravity and the position of the source in the `NominalFrame` and the distance between the impact point on the ground and the position in the telescope. h_{\max} is then calculated from the weighted average between

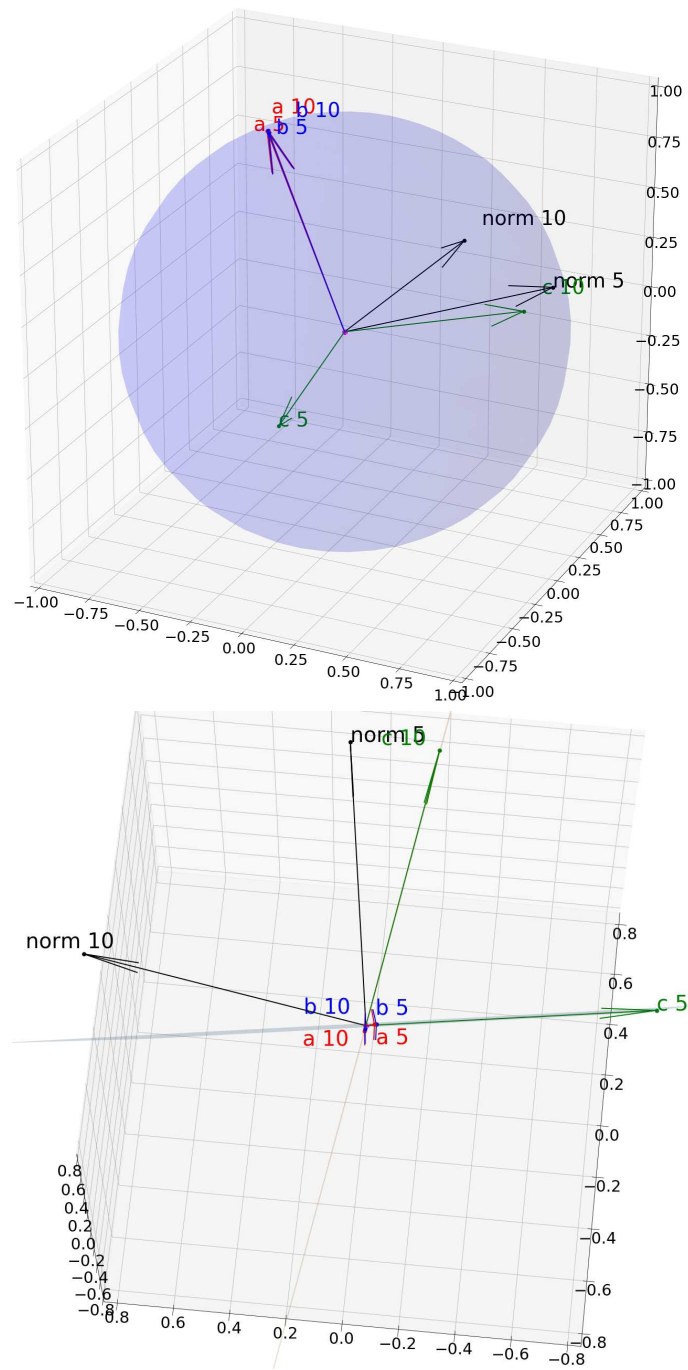


Figure 5.6: Plot done for MSTs with id 5 and 10. *Upper panel:* view of the unitary sphere used to display the vectors and points used for by the `HillasReconstructor`. *Lower panel:* this is a view from the top of the line intersecting the `HillasPlanes` the two telescopes. Note that the planes pass through **a**, **b** and **c** and **norm** is perpendicular to this plane.

the heights calculated by each telescope using as a weight the intensity of the ellipse, calculated in the hillas parametrization.

HillasReconstructor The `HillasReconstructor` uses a different approach with respect to the former reconstructor since the reconstruction of the incoming direction of

the primary particle producing the shower is performed in 3D. From the moments of the ellipse, two points are identified along the major axis, namely the center of gravity of the ellipse and another point along this axis (a value of 0.1 m from the c.o.g. has been chosen). These two points in the camera are projected in the sky, using the transformations between the frame of the camera, the telescope frame and the sky frame: this is like the reverse process of projecting the position of a star in the sky to see where it will be observed in the camera. The spherical coordinates of these two points are converted to the cartesian coordinates of two points on a sphere with a radius of 1. A plane passing through the two points on the sphere and the center of this 3D space is built for each telescope.

The planes are instances of the class `HillasPlane` and their attributes are the points on this unitary sphere (called **a** and **b**) and the point **c** belonging to the plane and the point norm, which is perpendicular to the plane (see figure 5.6): the last two points are output parameters calculated inside the class itself. Other two parameters are the position of the telescope and the weight used later for the reconstruction extracted from the hillas moments (calculated from `intensity * length / width3`). Those planes are then intersected pair-wise, and the angle between them is used as a weight for the computation of the final reconstructed direction, which is a weighed average between all pair-wise directions. The weights used in this last average are calculated multiplying the weight for each plane with the angle between these planes so that planes that are less parallel have a large impact on the final result.

The reconstruction of the impact point is similar to the one done in the `HillasIntersection` since it's also doing the intersection of the lines defined by the position of the telescope and the tilt angle of the ellipse (all is done in the `TiltedGroundFrame`).

This reconstruction of the impact point was adapted for the reconstruction of data simulated in the divergent pointing mode (see section 7.1.4). h_{\max} is also reconstructed from the 3D intersection of 3D lines, which are those defined by the vector **a** of the different `HillasPlanes`.

Energy estimation and classification The energy reconstruction in `ctape` is done with an `EnergyRegressor` class and an `EventClassifier` but it has not been fully implemented and tested yet.

Using the hillas moments and the telescopes positions with respect to the reconstructed impact point, the `EnergyRegressor` class has been used to get a first estimate for the energy, to be used as a seed for the ImPACT analysis. This could be done thanks to a LUT which was built using the hillas moments, the energy of the particle from the simulation and the distance of the telescope from the impact point: this LUT is created per each camera type and used taking the average of the energy estimation computed for each telescope.

³If only three pixels remains after the cleaning, these would result in a zero-width ellipse, with an infinite value for the weight. If this is the case, a warning is raised.

5.2.2 Model-based analysis

The analyses described until now are not exploiting the full camera images of an event. The camera images are parametrized as ellipses getting the Hillas parameters out of each camera image, that are later used for direction and energy estimation and for particle identification. This analysis technique was the only used for cameras with a low number of pixels, such as Whipple (109), CANGAROO (220) and HEGRA (271) in the early year of gamma-ray observation from the ground with IACTs.

Since the construction of the CAT imaging telescope, which was using a camera with 546 phototubes in a hexagonal matrix, a new analysis method was developed [95]. This method, which took advantage of the full imaging information, was based on the analytical modelling of the atmospheric showers induced by gamma-rays convoluted with the detector response. The consistency between the analytical calculations and the full Monte Carlo simulations have been checked, showing a good agreement for the lateral and longitudinal profiles of the gamma-ray images for various impact parameters.

The atmospheric shower was modelled considering many of parameters affecting its development through the atmosphere, such as the number of charged particles at a given atmospheric depth, their energy spectrum and the angular distribution of the momenta with respect to the shower axis. The average Cherenkov image can be deduced from additional informations, such as the atmospheric density profile, the optical absorption, the Cherenkov emission properties and some of the characteristics of the detector and of the observing site. At the end of the calculation, the distribution of Cherenkov light in the focal plane is a function of the gamma-ray energy, the distance between the telescope position and the shower axis, the source position and the angle between the shower image and a camera reference. A proper χ^2 -like function is minimised with respect to these parameters. Since the shower is modelled as a gamma-ray-induced one, the application of this modelling to a proton-induced shower will give bad results: this analysis can therefore provide gamma-hadron discrimination.

Model Analysis

A further development of this method has been later on adapted and used for the H.E.S.S. telescopes [96]. The cameras of the H.E.S.S. telescopes have a higher number of detectors with respect to previous instruments, with 960 photon detector elements for the smaller telescopes. This analysis, which explores not only the Hillas parametrization of the shower image but also the signal in the image, and it is used on a system of telescopes working in stereoscopy, increase the performances of the telescopes' system.

Model creation In the *Model Analysis* [96] the Cherenkov light distribution of a shower is determined by the lateral, longitudinal and angular distribution of charged particles in the shower, which are determined from Monte Carlo simulations and parametrized to provide a semi-analytical description of the shower, including the depth of first interac-

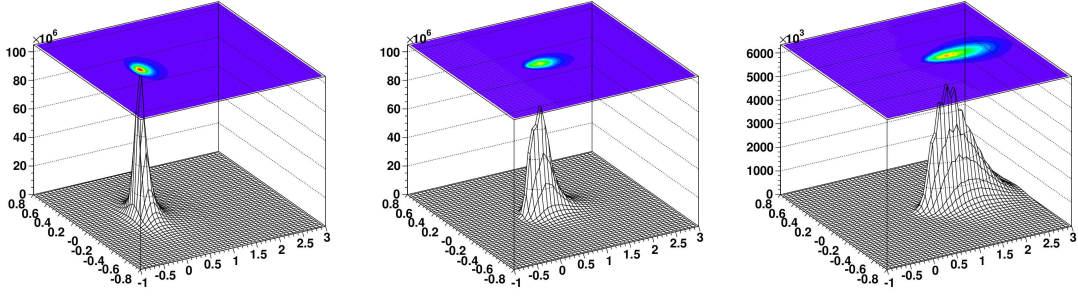


Figure 5.7: models for a 1 TeV shower started at one radiation length and falling 20 meters (left), 100 m (center) and 250 m (right) away from the telescope. The further the impact point is from the telescope, the more elongated the image in the camera is: note that the z-scale varies from 10^6 to 10^3 photoelectrons per bin, the x-scale and the y-scale are in degrees. Figures from [96].

tion as a new parameter with respect to the method used for CAT: this was done since this parameter is a source of shower-to-shower fluctuation while simulating the showers. The shower images are created convolving the Cherenkov information with the atmospheric absorption, the light collection efficiency, etc. The night sky background noise is also modelled on the basis of detailed statistical analysis. The parameters of the shower that best fit the observed camera images are calculated with a minimisation procedure which gives also a selection criteria to discriminate between gamma-ray induced signal from the hadronic background.

A template is produced per each zenith angle, impact distance, energy and interaction depth for a telescope pointing on-axis (the gamma-ray direction is at the center of the camera) and they are made of squared bins with a size of 0.01° (see figure 5.7). Once the templates are generated, they can be stored and used for the minimisation procedure in which the intensities of the pixels in the camera are compared with the prediction from the templates: the prediction is created by interpolating between the templates associated to the grid points closest to the actual set of parameters (see the ImPACT session for further explanation).

Likelihood If we consider a single telescope, at this point we have a camera formed by N pixels, with signal s_k in each pixel, and the prediction, with signal μ_k . The likelihood $P_k(s_k|\mu_k, \sigma_p, \sigma_\gamma)$ to observe the signal s_k (which is in photo-electrons) in a pixel for an expectation value of μ_k is given by the convolution of the Poisson distribution of the photo-electron number n (the number of observed photoelectrons) with the photo-multiplier resolution. The latter can be approximated by a Gaussian having a width of $\sqrt{\sigma_p^2 + n\sigma_\gamma^2}$, where σ_p is the width of the pedestal (width of the histogram of the charge in the pixel outside the signal region, including night sky background) and σ_γ is the width of the single photo-electron peak (namely the photomultiplier resolution). The likelihood

$P_k(s_k|\mu_k, \sigma_p, \sigma_\gamma)$ per each pixel is then given by:

$$P_k(s_k|\mu_k, \sigma_p, \sigma_\gamma) = \sum_n \frac{\mu^n e^{-\mu}}{n!} \frac{1}{\sqrt{2\pi(\sigma_p^2 + n\sigma_\gamma^2)}} \exp\left(-\frac{(s_k - n)^2}{2(\sigma_p^2 + n\sigma_\gamma^2)}\right). \quad (5.1)$$

The cumulative likelihood for the whole camera image being a sum over the P_k , the pixels passing the cleaning. The total likelihood for an event is the sum over all the cameras performing the event reconstruction. The underlying consideration when summing over all pixels is that the pixel are treated as independent and they can also have different values for the pedestal σ_p . The knowledge of both σ_p and σ_γ during an observing run in a real scenario, make sure that the algorithm is robust against varying levels of night sky background.

In case of high expected values in the pixels (meaning $\mu \gg 0$), the Poisson distribution can be replaced by a Gaussian of width $\sqrt{\mu}$ and P_k becomes:

$$P_k(s_k|\mu_k \gg 0, \sigma_p, \sigma_\gamma) \approx \frac{1}{\sqrt{2\pi(\sigma_p^2 + \mu(1 + \sigma_\gamma^2))}} \exp\left(-\frac{(s - \mu)^2}{2(\sigma_p^2 + \mu(1 + \sigma_\gamma^2))}\right), \quad (5.2)$$

which will be again calculated per each pixel and summed over all the pixels in all the cameras. The pixel used for this calculations are only those passing the cleaning, since the othe

In order to have a variable which behaves asymptotically as χ^2 , the *pixel log-likelihood*, defined as $\ln L = -2 \ln P_k(s_k|\mu_k, \sigma_p, \sigma_\gamma)$ is used. This function here can be averaged integrating over s and in the limit of $\mu \gg 0$ it takes the form:

$$\langle \ln L \rangle_\mu = 1 + \ln(2\pi) + \ln(\sigma_p^2 + \mu(1 + \sigma_\gamma^2)), \quad \sigma^2(\ln L) = 2. \quad (5.3)$$

The log-likelihood for the whole telescope will simply be the sum over the *pixel log-likelihood*.

Goodness-of-fit In order to compare the model prediction and the actual shower images and discriminate between gamma-rays and hadrons, a goodness-of-fit approach is used, defined as:

$$G = \frac{\sum_{i=1}^{N_{\text{pix}}} [\ln L(s_i|\mu_i) - \langle \ln L \rangle_{\mu_i}]}{\sqrt{2 \times \text{NdF}}}, \quad (5.4)$$

which is a normalized sum between the actual pixel likelihood that we get from computing the logarithm of eq. 5.2 and the expected average value calculated with eq. 5.3 using as μ the signal μ_i in each pixel of the predicted template. NdF is the number of degrees of freedom which is the number of pixels in the camera, minus the 6 shower parameters (2 for direction, 2 for impact parameter, depth of first interaction and energy). The goodness-of-fit behaves asymptotically as χ_{NdF}^2 (where **NdF** is the number of degrees of freedom) and

can be used both to provide a measure of the fit quality and to be used for gamma-hadron separation since the templates are created under the hypothesis of a gamma-ray and a hadron-induced shower will result in a much worse goodness-of-fit.

3D Model

In the 3D Model Analysis [97] the shower is modelled as a Gaussian ellipsoid in the atmosphere, which is then integrated with a path integral along the line-of-sight to predict the light collected in each pixel. A log-likelihood is used to compare the actual image with the predicted one to reconstruct the shower parameters, such as mean altitude, impact point, direction, 3d width and length and luminosity. Energy reconstruction and gamma-hadron separation are also possible with this analysis.

5.2.3 Template-based analysis: ImPACT

A similar analysis with respect to the one presented in the Model Analysis (sec. 5.2.2) is called ImPACT (Image Pixel-wise fit for Atmospheric Cherenkov Telescopes) [98]; this algorithm is also based on the creation of a library of templates that are used in order to interpolate the camera images between them to get the shower parameters, with the main difference that those templates don't come from a semi-analytical calculation, but are instead created from *ad-hoc* Monte Carlo simulations. CORSIKA and sim_telarray, the same softwares used also in CTA for the Monte Carlo production, have been used for the creation of the templates.

The fitting procedure and the general approach is identical to the Model Analysis apart from the gamma-hadron separation part, which was not performed using the goodness-of-fit but with boosted decision trees based method already implemented in the H.E.S.S. analysis framework; the decision of not using the goodness-of-fit in this process relies on the strong dependence of this value on the NSB level and on the good knowledge of the single photoelectron response and pedestal width which might not be very stable between different observations conditions.

ImPACT is an event reconstruction algorithm which was developed for the H.E.S.S. experiment and has reduced the time required to perform the detection of a point source by a factor of 2 when compared to the standard Hillas-based reconstruction (bottom plot in figure 5.8). The improvement in sensitivity, and therefore the lower time required to get a 5 sigma detection, comes from the smaller θ^2 (square of the angle between the reconstructed position and the source position) cut allowed by the improved PSF. The effect of a harder cut which can be applied in the ImPACT analysis is to reduce the rate of cosmic ray background, while the effective area remains roughly the same. The energy and angular resolution also improves thanks to the ImPACT analysis which means that morphological studies of extended sources benefit from this analysis (see figures 5.8 from [98]).

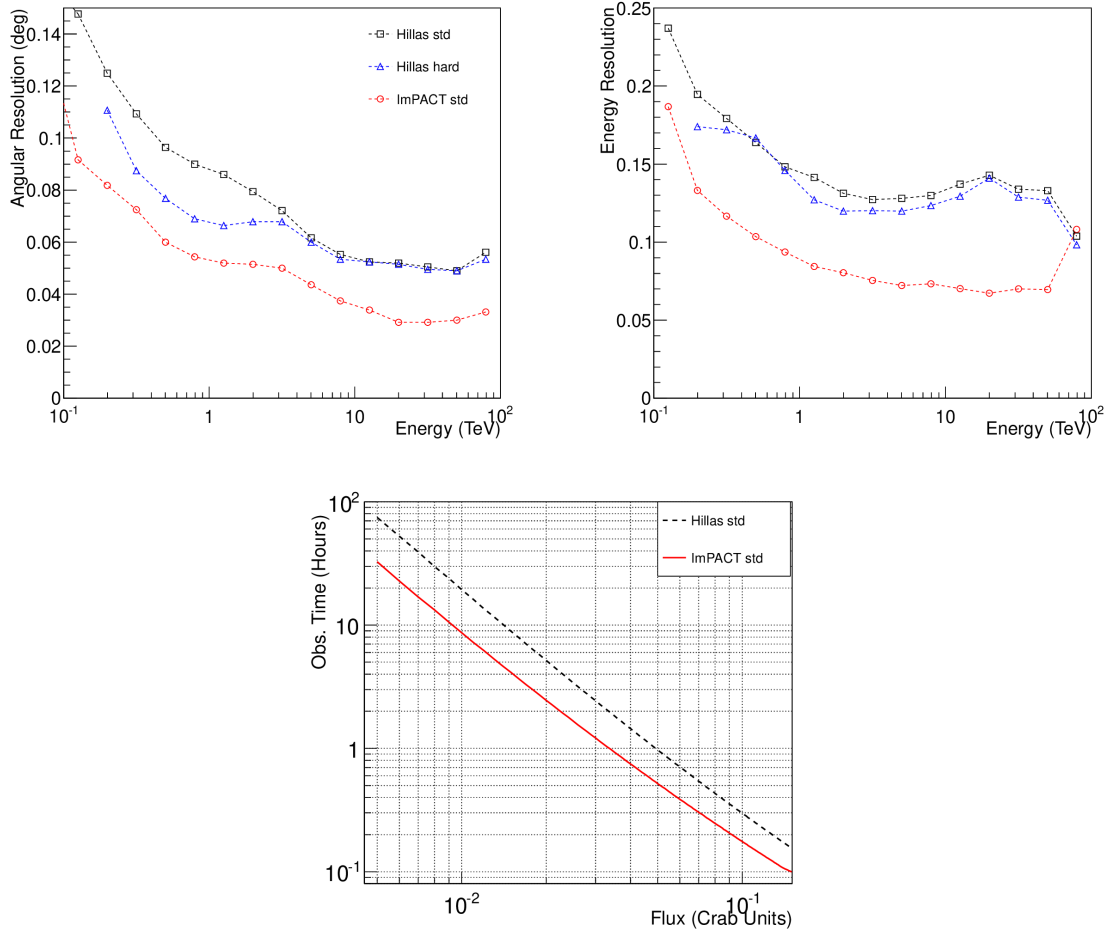


Figure 5.8: angular resolution (upper left) and energy resolution (upper right) of the ImPACT analysis compared to two different standard Hillas analysis with standard cuts and hard cuts. The plot in the lower panel represents the observation time require to detect a point-like source with a statistical significance of 5σ as a function of the flux of the source (same spectrum of the Crab Nebula). Plots from [98].

ImPACT in ctapipe

The ImPACT analysis in the `ctapipe` framework has been implemented mainly by Dr. Daniel Robert Parsons, whom I have been working with during the PhD and especially during the research period spent at LAPP. I started to use the ImPACT analysis thanks to some scripts, the templates and the code provided by Dr. Parsons: after making sure that the analysis was working properly, I started to profile the script in order to understand which parts could be changed in order to save computing time.

The goal of the work was to debug the analysis, do the time profiling the pipeline to identify the bottlenecks and to fully understand the code in order to look for those calculations which could take advantage of the computing power of a Graphics Processing Units: this work will be presented in chapter 6.

The generation of the templates has been done by Dr. Parsons whereas I contributed

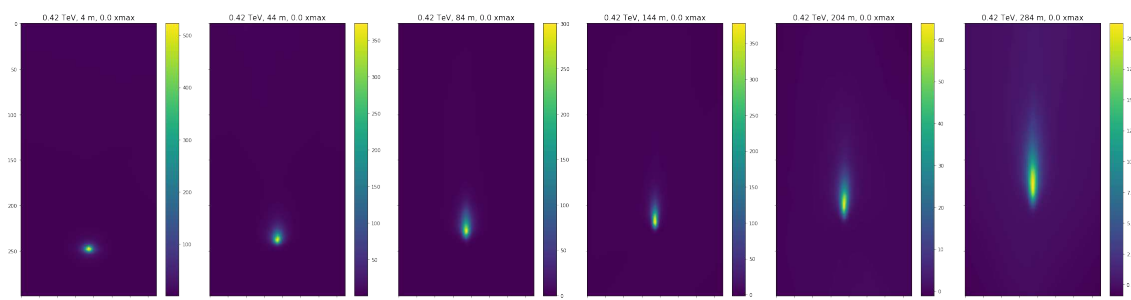


Figure 5.9: 6 selected templates for the LST camera for a shower at 420 GeV hitting the ground at 4, 44, 84, 144, 204 and 284 meters from the telescope. The value of the scaled x_{\max} parameter is 0, which means that the height at which the events used to build those templates are reaching the maximum emission fall in an range centred at the value given by equation 5.5 and 25 g/cm^2 large (see text).

to the testing and debugging of the ImPACT code for CTA and also to the analysis of the full array configuration, the LST-only subarray and later on adapted the scripts for the LST-1 prototype telescope working in “mono mode”.

Templates creation The first step in the ImPACT analysis is the creation of the templates, which is done from *ad-hoc* Monte Carlo simulations using CORSIKA and `sim_telarray` : those have been created at first using a tweaked version of `sim_telarray` in order to trace the incoming photons through the telescope optics in the camera frame in a finely binned histogram after applying the telescope and detector efficiencies.

Recently the creation of the templates has changed⁴ and is using the standard version of the `sim_telarray` software together with a different fitting scheme. For the creation of the templates per each camera type, a telescope is placed on the ground at different positions with respect to the impact point of the shower (as it was done for the previous case), with a spacing of 40 meters between each telescope. The direction of the simulated particles is not a point source but a diffuse source with a small view-cone of 0.5° and the simulation and calibration is done in the same way as in regular Monte Carlo productions. In order to create the templates in the camera, a neural network is trained to reproduce the final template, combining the images obtained at the same energy, impact point and x_{\max} bin. The images are coming from a slightly different region in the sky because the simulated gamma rays are not coming from a point source but from a diffuse source (with a small view-cone angle).

A library of templates is created per each camera type and for each value of energy, impact point and x_{\max} there is a template: a total of 7043 templates for the camera of the LST are created for 21 log-spaced values for the energy (17 GeV – 167 TeV), 24 linearly spaced values for the impact point (4 m – 504 m) and 16 x_{\max} bins (see example templates in figure 5.9)⁵. The x_{\max} value is saved as the difference between itself and the

⁴See https://github.com/ParsonsRD/template_builder for more details.

⁵There are not 8064 templates, as one might expect from the binning used for the three parameters. The missing templates are those characterized by values of the parameters at the edges of their ranges (i.e. there are no templates for low energy and high impact point). The edges of this parameter space are not populated.

average value which is expected considering the energy of the incoming particle (adapted from equation 3 in [96]),

$$x_{max} = 300 + 93 \log \left(\frac{E}{1 \text{ TeV}} \right), \quad (5.5)$$

so that x_{max} ranges from -125 and 250 g/cm² (with a bin size of 25 g/cm²). This means that the events used to produce a template with energy E_k , distance of impact point d_k and at x_{max_k} have been simulated at a specific energy, placing the telescope at a specific distance from the impact point and only those events falling in the same x_{max} bin (centered at x_{max_k} and 25 g/cm² large) are used to create one template.

The size of those templates is 150 × 300 pixels, with a pixel size of 0.02°. Each template will be labelled with its three characteristic values. Technically speaking, the template library is read as a Python dictionary, with the keys being the template label and the value being the 2D template itself.

The implementation of a pipeline and of the code that will be described in the next sections is referred to the version 0.7 of `ctapipe`.

Full pipeline A full standard pipeline, including data loading, calibration and cleaning, is presented in section 5.1.2.

The `ImPACT` code can be run with three main steps: initialization of an instance of the `ImPACTReconstructor` class specifying the path of the templates together with the minimiser to use and some other parameters used to tune the analysis. Then the useful data from one event are loaded, such as the coordinates of the telescopes, the camera images and Hillas moments: those data are given using the `set_event_properties` method as Python dictionaries, using the telescope ID as a key. While the Hillas parameters are usually calculated in the frame of the camera and parameters such as the position of the ellipse are expressed in meters, the `ImPACT` analysis performs the reconstruction in the `NominalFrame`. This is a common frame between the telescopes that are used in the event reconstruction and its basically a spherical frame, where the distances between points are measured with angles. In order to go from the frame of the camera to the `NominalFrame`, the Hillas parameters must be provided in radians dividing the values computed in the frame of the camera for the focal length of the telescope.

Finally the `predict` is called: this method needs as input seeds some shower parameters, like impact point and direction, previously reconstructed with a `Reconstructor` and the energy estimated from some external look-up table previously filled from Monte Carlo simulations: the LUT is then used to estimate the energy of a shower (per telescope) using the Hillas moments in each camera and the distance between the telescope and the impact point on the ground to estimate the energy. The final estimate, which will be a seed for the `ImPACT` analysis, will be an average between the single estimate from each telescope.

“Mono mode” for LST-1 As already said, the ImPACT analysis needs three seeds to be done, which are a first guess on the estimate of the reconstruction of the impact point on the ground, the x_{\max} parameter and the energy of the event. While these three parameters can be calculated from the stereoscopic reconstruction of an event, this is not true any more in the case of an observation carried one with just one telescope, such as the first prototype of the LST telescope which is in commissioning phase in La Palma.

Together with the work being carried on for the `ctapipe` software package, a library called `cta-lstchain` (<https://github.com/cta-observatory/cta-lstchain>) built on top of the `ctapipe` environment is being developed for the analysis of the data of LST-1. This library uses most of the methods and classes from `ctapipe`, together with other scripts that are used to do the full reconstruction in “mono mode” included low-level scripts to read real data from the LST prototype.

The goal of my analysis was to use ImPACT to work on the performances of LST-1 in mono mode and the first step has been the refactoring of the scripts that I was using to analyse the data with an array of telescopes (such as the full array for Paranal or only the 4 LSTs in La Palma) in order to use them for the data analysis of the LST-1. In order to make sure that the script was working properly and the format of the input data was correct, I used the Monte Carlo informations to create the input data for the analysis. The energy was taken directly from the MC simulation, the distance between the impact point and the telescope position was computed from the impact point in the simulation and a simple x_{\max} was estimated from the energy of the primary particle using eq. 5.5. Once I was sure that the mono analysis could be done in ImPACT, I focused my attention on the implementation of an algorithm which could generate the input for ImPACT without using the informations in the Monte Carlo simulation.

The method that I came up with uses the Hillas ellipse to draw a line which extends the major axis of the ellipse: from this we know that if we project the incoming direction of the particle from the sky to the camera frame, we obtain a point in the camera frame which should fall in this line given by the major axis. In this analysis I’m doing the reverse process, looping over some points along this line in the camera frame and projecting each point in the sky to get an incoming direction of the particle each time (see figure 5.10).

This is valid if the analysis is being carried on diffuse gamma rays, while it can be simplified if one assumes that the source is expected to be in a certain point in the camera (source-dependent analysis), so that this parameter can be kept fixed. If, for example, we are analysing gamma rays coming from a point source which is falling at the centre of the camera, the seed for the incoming direction of the gamma rays can be chosen to be the pointing direction of the telescope. At this point we have a value (or a set of values) for the incoming direction, one of the three parameters needed by ImPACT.

The reconstruction of the impact point is performed in the `TiltedGroundFrame` drawing a line passing through each telescope position and having a slope given by the rotation angle of the ellipse observed in the camera: these lines are then intersected to find the impact point in this plane, which is then projected in the `GroundFrame`. The intersec-

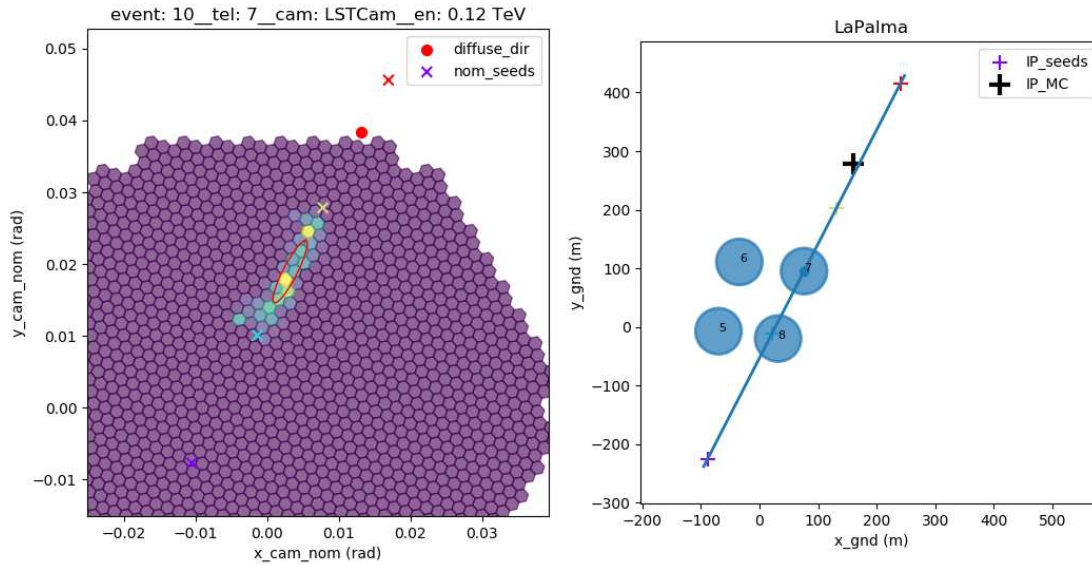


Figure 5.10: Representation of the analysis of gamma rays coming from a big region of the sky (diffuse gamma rays). In the camera image on the left the ellipse is drawn in red, the crosses are drawn along the major axis of the ellipse and incoming particle direction is also represented as a red circle. The image on the right shows the LST which is being analysed (7) and the line drawn together with the seeds for the impact point (coloured “plus” markers) and the impact point from the Monte Carlo simulation.

tion of multiple lines is not possible with just one telescope but if the cleaning and the parametrization are done properly, the line drawn from the telescope observing in “mono mode” should point to the position of the impact point. It’s important to do the cleaning properly, otherwise a wrong calculation of the Hillas parameters will give worst results at the end. The seeds for the impact point are chosen on this line with a spacing of roughly 30 meters. Now we have a set of values for the incoming direction of the particle and for the impact point and we only miss the energy.

As mentioned in the previous paragraph, the moments of the ellipse and the position for each telescope are used to get an estimate of the energy of the primary particle using a LUT and then taking the average over the estimate from each telescope. The estimate from a single telescope can be done also in the case of the observation in “mono mode” and even though the precision in the estimation of the energy seed is not really great, it can be good enough to be used as a seed by ImPACT. Every time the seed position for the impact point will be different, there will be a new value for the energy which is also going to be used to determine the value of x_{\max} .

Once we have the set of possible incoming directions, impact points and energies, we can loop over the different set of seeds and get the reconstructed shower parameters for each new set of input seeds. The final result which is going to be considered as the best one, after all the sets of seeds have been tested, will be the one with the lower value for the goodness-of-fit parameter.

The angular and energy resolutions obtained analysing few hundreds events are re-

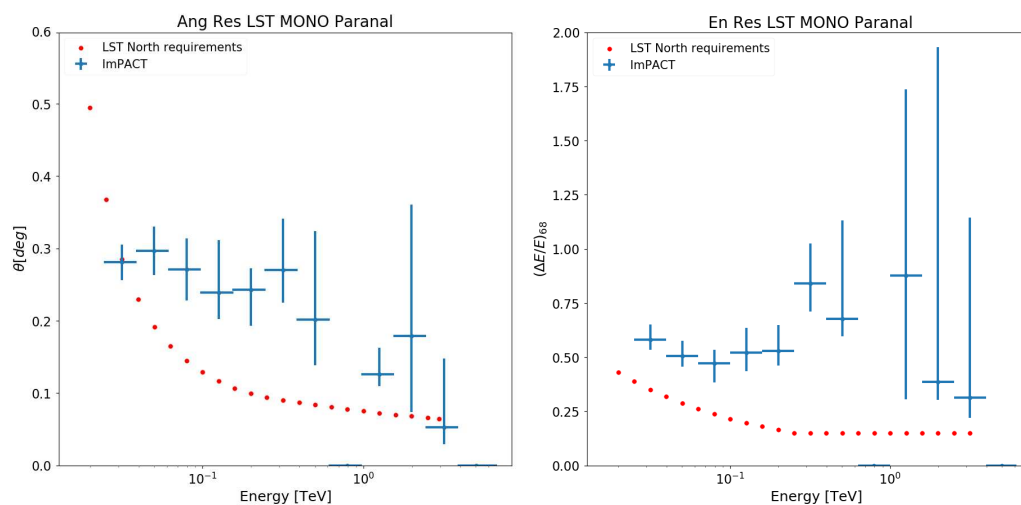


Figure 5.11: Angular resolution and energy resolution obtained analysing 741 events for a single LST in Paranal plotted against the requirements for the 4 LSTs. The high energy part, from roughly 800 GeV is missing statistics and therefore the points are either not plotted or have a large error bar.

ported in figure 5.11. The result for the angular resolution, even if statistics is low, is quite good since the reference they are compared with is the angular resolution for the array of 4 LSTs, which is not a fair comparison since there's a huge difference in using 4 telescope or using just one telescope.

This is a first implementation of the ImPACT analysis for the LST-1 and there are many optimizations that must be done. A really useful one would be to use the output from the `lstchain` analysis (the `ctapipe`-based pipeline being implemented for the analysis of LST-1 data) as a seed for the ImPACT analysis with LST-1, since a big drawback of this looping N times ($N = 10$ in these analyses) over different seeds results in an analysis which is roughly N times slower with respect to the stereoscopic one. The output from `lstchain` would be used for the ImPACT analysis in mono mode with no need for looping over several initial seeds.

The ImPACT code

Initialization During the initialization of the class, which is done once at the beginning the analysis, the paths of the template libraries for each camera is loaded into memory (each camera template will be loaded only if needed) and the minimiser is chosen together with some other scaling parameters. The `set_event_properties` method is then used to load the camera images, telescopes positions, pixels positions and all the other necessary data and finally the `predict` method is called: this is responsible of preparing the inputs for the minimiser and starting the minimisation procedure.

From the inputs given, each parameter which is going to be used in the minimiser will have its initial value, step and limits that the minimiser is going to use in the minimisation procedure.

Minimiser The input parameters, the steps and the limits, are the inputs to the `minimise` method of the `ImPACT` class which behaves slightly differently according to the minimiser which is chosen between those implemented, such as `minuit`⁶, `nlopt`⁷ in its Python implementation and the least squares method from the Scipy library.

Regardless of the choice of the minimiser, the function `get_likelihood` is called at each iteration of the minimisation procedure and its outputs are used to move in the parameter space: the likelihood for each camera image is summed between all the telescopes and it is used by the minimiser to move in the parameter space.

The Interpolator class In order to compute the likelihood, a comparison between the camera image and the templates must be done: at this stage the interpolation class is used. The comparison between the camera images and the templates is done via the `TemplateNetworkInterpolator` class which calls the `UnstructuredInterpolator` class. The idea behind this last class is to create a 3D parameter space, with the three dimensions being energy, impact point and xmax, and to connect the n -th template to its representative point $P_n(E, \text{impact_point}, \text{xmax})$ in this 3D space given by the template's label itself: for each point in this parameter space, there will be an associated template.

During the i -th step in the minimisation routine, each telescope will have a set of three trial values (energy, impact point and xmax) which is a point P_{k_i} in this 3D space (each k_i is the label of a point in this space and i is the label for each telescope in the event). In order to get a prediction from the templates to be compared with the camera image, we consider the points in the 3D space which are closer to the trial points P_{k_i} . In an event in which, for example, 5 telescopes are being analysed (i ranges from 1 to 5), 20 templates will be used, 4 per each telescope (the reason for this number is explained in the next paragraph). The templates associated to those points, 4 per each point, are expected to be those best representing the camera image in each telescope at that certain step i .

Delauney triangulation The 3D space (one per telescope type involved in the analysis) is created just once at the beginning of the analysis using the label of each template, and the subsequent localisation of the closest points is done by means of a Delauney triangulation. The space is divided into simplices, namely tetrahedrons⁸, whose vertices are nothing but the points representing the templates, and they are created so that no point of the triangulation is falling inside any tetrahedron. An example of a 2D Delauney triangulation is reported in figure 5.12.

At each iteration of the minimisation algorithm, each trial point (one per telescope) belonging to this 3D space will fall inside a tetrahedron with 4 vertices (meaning that there are 4 templates that are the closest to the trial point) and from the points defining the vertices of the tetrahedron we immediately know which are the associated templates.

⁶<https://iminuit.readthedocs.io/en/latest/>

⁷<https://nlopt.readthedocs.io/en/latest/>

⁸The general theory of the Delauney triangulation states that in a space with d dimensions, the simplices will have $d + 1$ vertices, so that triangles will form in a 2D space and tetrahedrons will form in 3D.

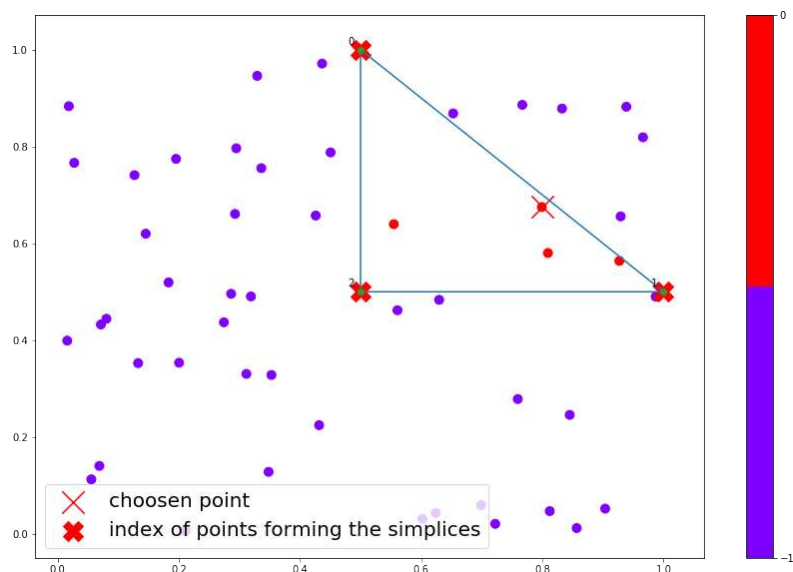


Figure 5.12: An example of Delaunay triangulation in 2D. The figure is produced creating a triangle (the only simplex is this space) and then placing 50 random points inside the triangle (the trial points). In this figure the red crosses in bold represent the points characterising the templates in the ImPACT analysis. The points are the result of the fitting routine and they are used to show that each point knows if it is placed inside the triangle (triangle number is 0) or outside of it (triangle number is -1).

Interpolation The last piece in the interpolation in the `UnstructuredInterpolator` class is the calculation of a predicted image corresponding to one of the vertices: the idea is to map the coordinates of the pixels passing the cleaning with the proper pixels in the template. Since the templates are simply images without physical coordinates, the first step is to assign a physical coordinate to each pixel in the template and then to map the positions of the pixel in the camera frame to those in the template. Each pixel in the camera frame is larger than the pixels in the template since the size is 0.1° for the former and 0.02° for the latter. More details regarding this will be explained in the PyTorch implementation of this routine in chapter 6.

The final prediction is calculated with a weighted average over the 4 closest templates, using as weights the distances between the test point P_{k_i} and the 4 vertices.

After the prediction (one per each telescope) has been created from the 4 surrounding templates, it is compared with the respective camera image in order to compute the likelihood that the prediction matches with the image in the camera: at each step the likelihood is computed (using the eq. 5.2) and used as a parameter for the convergence of the minimiser.

5.2.4 Machine Learning

Many groups are testing different implementation of an energy estimator, using machine learning techniques for training neural networks on Monte Carlo data and using them to test the goodness of the reconstruction.

This is being tested in **protopipe**⁹, the prototype of the pipeline for the low-level data analysis of the CTA data, built on top of **ctapipe**. There's also a group of colleagues from the FACT team who have implemented the **aict-tools** <https://github.com/fact-project/aict-tools> to perform energy regression and gamma/hadron separation in FACT and in CTA.

The idea behind these tools is to use the Monte Carlo informations together with hillas parameters to train a model which is then used with other Monte Carlo simulations, or on real data, to estimate the energy of the incoming particle based on the trained model. This is done for each type of telescope. Similarly, the gamma/hadron separation is performed with a model trained on gamma rays and protons and later on this model is applied to another set of Monte Carlo simulation simulations for testing.

These machine learning techniques are similar to those applied to the analysis with the MAGIC software, which uses Random Forests for energy estimation and event classification (see section 1.3 on the IRF production in CTA).

Deep Learning

Together with the more classical machine learning methods that are nowadays used for the event reconstruction in all the existing IACT systems, Deep Learning is also being investigated due to the many fields in which it has been applied in the last years, such as the natural language processing and the computer vision.

This method is a class of machine learning algorithms which ingest a raw level input, which is passed through a series of layers, each of them extracting a higher level feature from the input data.

GammaLearn [99] is one of the projects which is being developed for the analysis of IACT data, particularly for those from CTA. The goal of this framework is to find the best possible neural network, using different layers connected in different ways to form a network, with the goal of achieving the best gamma/hadron separation and the best reconstruction of the parameters of the gamma ray. Together with this framework, there are other groups in the CTA consortium working on the development of Deep Learning methods for the data analysis of CTA data and many others have been developed and used in other IACTs experiments, showing how a particular type of deep neural network called Convolutional Neural Networks (CNNs) can be used for these type of analyses (see references in [99]).

While the machine learning algorithms implemented in a IACT analysis pipeline are using the hillas parameters, the Deep Learning methods are using the DL1 images before any cleaning or hillas parametrization is applied, and the deep neural network is supposed to learn to extract the important features of the image by itself.

⁹<https://github.com/cta-observatory/protopipe>

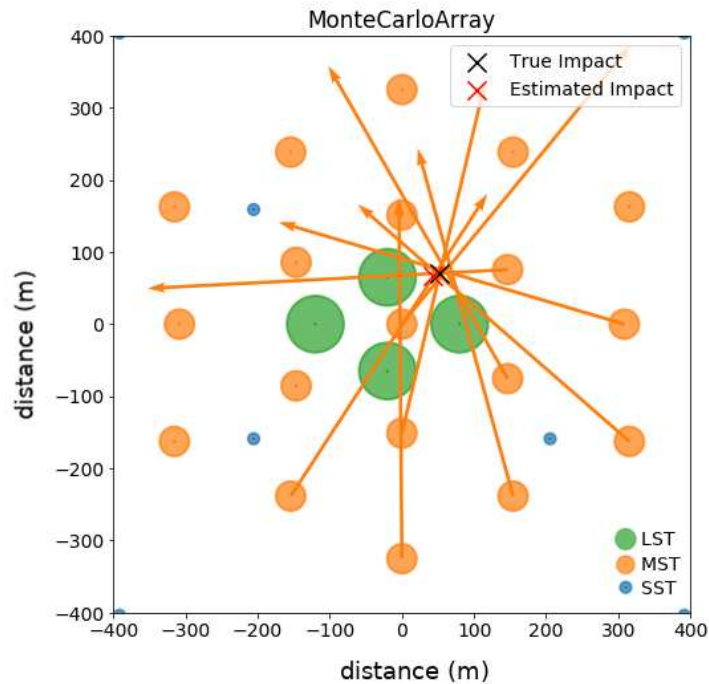


Figure 5.13: figure created with the `ArrayDisplay` showing the positions of the telescopes in the `GroundFrame`, the MC and reconstructed impact point and the arrows for every telescope participating to the reconstruction showing how each telescope participates to the reconstruction of the impact point. Figure from the tutorial in the `ctapipe` documentation.

5.3 Visualization

The visualization of the event reconstruction is a really important feature in `ctapipe` since it allows to visualize the goodness of the reconstruction. The `ArrayDisplay` in the `ctapipe.visualization` allows to plot the telescope positions either in the `GroundFrame` (see figure 5.13) or in the `TiltedGroundFrame`, the latter being particularly useful for the visualization of the reconstruction which is done in this frame.

5.3.1 OpenSCAD displayer

The first implementation of the 3D displayer has been done using SolidPython¹⁰ to generate an OpenSCAD¹¹ which could be then rendered. From this first implementation, not easy to implement due to the lack of documentation, I managed to produce some 3D plots and to understand how the plotting of the camera image in `ctapipe` is related to the real position of the camera in the telescope.

There are two types of plots that can be done using this viewer: the first one is the visualization of the event itself with the image in the cameras, a mock structure of the

¹⁰<https://github.com/SolidCode/SolidPython>

¹¹<https://www.openscad.org/>

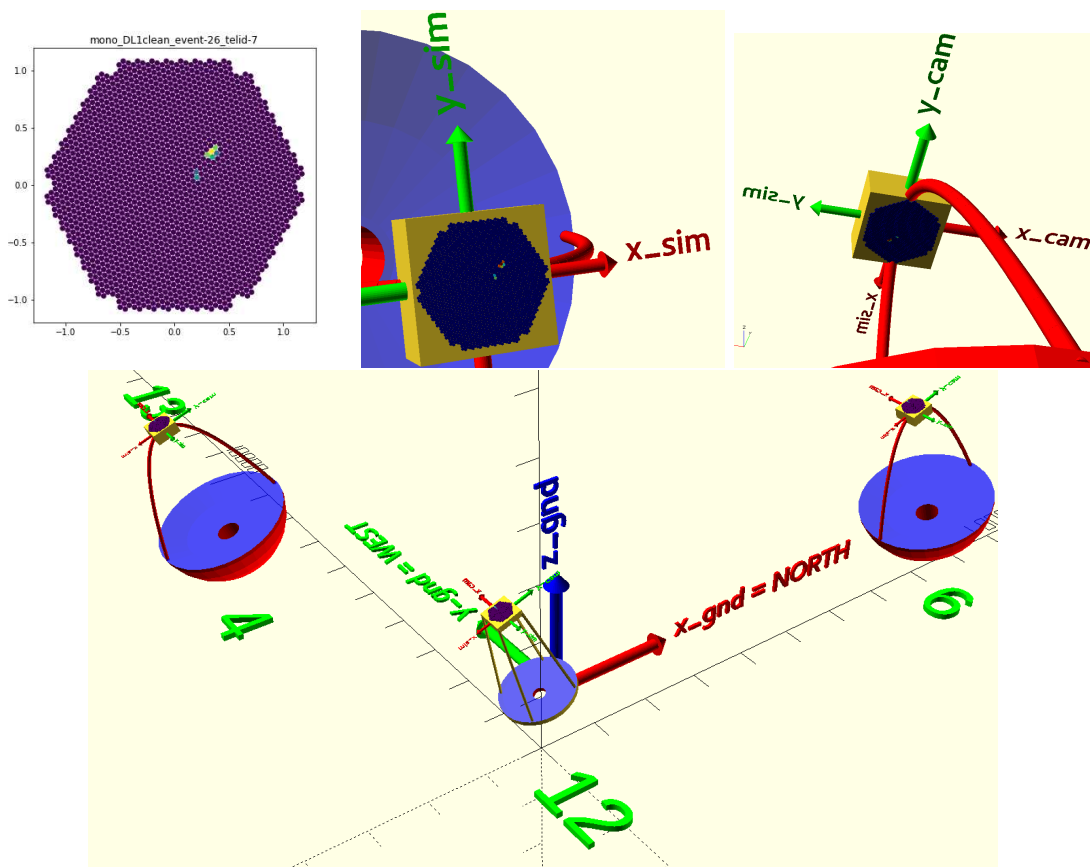


Figure 5.14: These plots shows the output of `ctapipe` compared with the one from the 3D displayer in OpenSCAD. Upper panel: the image from the `ctapipe` camera displayer showing the DL1 image of an LST is compared with the 3D one (x_{sim} and y_{sim} axis). The x_{cam} and y_{cam} axis) represent what one would see from the mirror and shows how this is different from the `ctapipe` displayer. Lowe panel: a display from another event showing three telescopes (both LSTs and MSTs), the reference frames in the cameras, the IDs on the ground and the GroundFrame orientation related with the cardinal directions.

telescopes, its ID and the various arrows to show the various frames that are being used (see figure 5.14). Thanks to this plot I understood that the real camera image created with `ctapipe` and the real camera image obtained observing the camera from the mirrors are connected one with the other with the transformation $x_{cam} = -y_{sim}$ and $y_{cam} = -x_{sim}$ (see figure 5.14).

The second type of plot shows the telescopes as spheres in the `GroundFrame` and in the `TiltedGroundFrame` (faster to render than the telescopes with their cameras) to understand how the positions of the telescopes in these two frames are related one another and where is the North direction (see figure 5.15).

5.3.2 CREED_VTK

One issue with the previous implementation was the speed of execution when trying to plot more than few telescopes since the `.scad` has to be created at first and then rendered. The second main issue with this displayer was its non usability by any other person, since

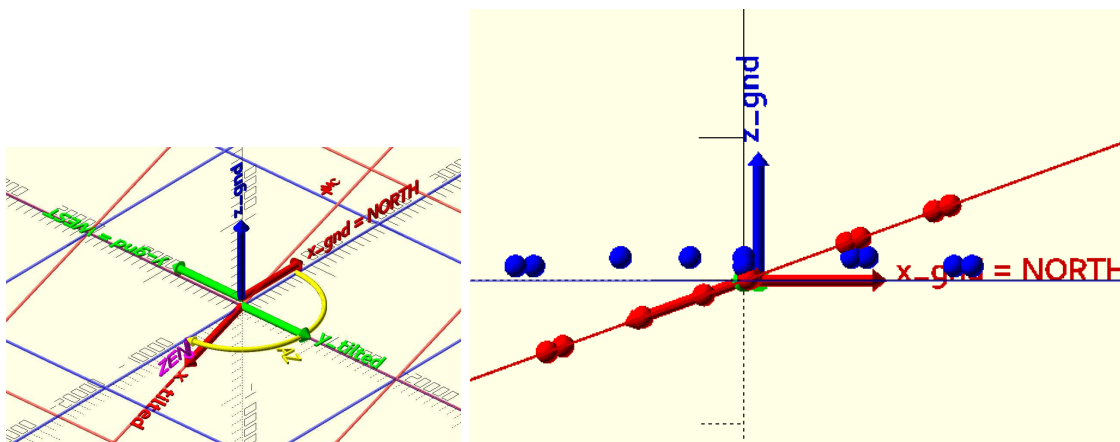


Figure 5.15: representation of the **GroundFrame** and the **TiltedGroundFrame**. In the left figure, both frames are represented and they show how the coordinates specified in one are related with those specified in the other, while right plot (side view) it's easy to see that the positions in the tilted frame are just a projection of those in the ground frame.

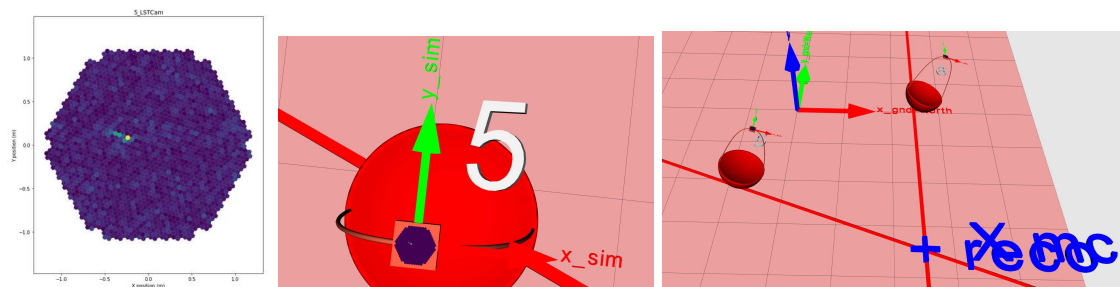


Figure 5.16: the image on the has been created with **ctapipe**, the central image is a zoom on the same telescope from **CREED_VTK** showing the line on the **GroundFrame** and the image on the left is the event with the two telescopes used in the reconstruction. The impact point from the MonteCarlo (**X mc**) and the reconstructed one (**+ reco**) are reported.

the code was both doing the analysis and producing the output, and each time a different visualization has to be done, the **ctapipe** analysis has to be done again.

I then decided to refactor the 3D displayer using a very powerful library called VTK (Virtual Toolkit)¹², an open-source software for 3D computer graphics, modelling, image processing and scientific visualization. This new displayer is reported in the **ctapipe** documentation as an optional package.

The two issues related to the speed and the usability have been solved with this software since I made a proper Python package called **CREED_VTK**, which can be installed in the **ctapipe** conda environment from the GitHub repository (https://github.com/thomasgas/CREED_VTK) as any other Python package. This displayer can be used in any pipeline and it creates the same output of the previous version with OpenSCAD, namely the telescope structure with the cameras (figure 5.16) and the event visualization, and the plotting of the positions of the telescopes (figure 5.17) with the ground frames.

The code used for the creation of the 3D images of the telescopes with the event

¹²<https://vtk.org/>

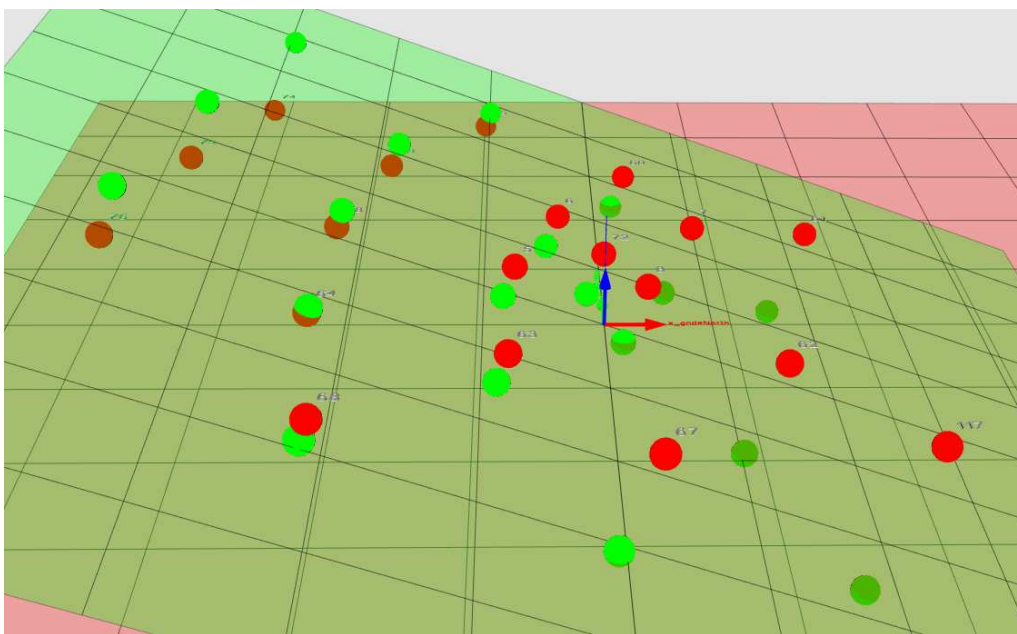


Figure 5.17: telescopes in the `GroundFrame` and in the `TiltedGroundFrame` plotted together with the two planes using `CREED_VTK`.

reported in the camera is:

```

from CREED_VTK import CREED_VTK
...
render = CREED_VTK(event, telescopes_ids=list(hillas_dict.keys()))
render.event_type(clean_level = "clean", clean_dict=cleaned_dict)
render.add_arrows_camera_frame()
render.add_gnd_frame(size=1000)
render.camera_view(elev=20)
render.tel_labels()

gnd_reco_pos = GroundFrame(x= reco.core_x, y = reco.core_y, z = 0 * u.m)

render.add_impact_point(label="X mc", status="mc", frame="ground")
render.add_impact_point(status="reco",
                        label="+ reco",
                        gnd_reco_pos=gnd_reco_pos,
                        frame="ground")

render.plot_hillas_lines(hillas_dict=hillas_dict, length=500, frame="ground")

render.show(width= 1600, height=1000)

```

where `hillas_dict` is a dictionary with all the hillas moments, `cleaned_dict` is another dictionary with the cleaned DL1 images, `reco` is the output from the `HillasReconstructor`. Changing the frame from `ground` to `tilted` will display the reconstructed impact point

in the `TiltedGroundFrame`.

The code used for creating the image of the telescope in the ground frame is:

```

from CREED_VTK import CREED_VTK
...
render = CREED_VTK(event, telescopes_ids=layout)
render.add_gnd_tels()
render.add_gnd_frame(size=1000)
render.add_tilted_frame(size=1000)
render.add_tilted_tels()
render.camera_view(elev=20)
render.tel_labels()
render.show(width= 1600, height=1000)

```

where the only used parameter is `layout`, which is the list of telescopes IDs to be shown in the displayer.

Thanks to the VTK library, the speed issue has also been solved since the displayer can also plot an event in which tens of telescopes have been triggered, each of them with the event visualized in the camera, in a few seconds.

An important addition to this 3D displayer with respect to the previous one, making this similar to the `ArrayDisplay` class in `ctapipe`, is the plotting of the lines coming from the hillas parametrization in both the `GroundFrame` and in the `TiltedGroundFrame` (see figure 5.18).

This 3D displayer was quite useful for debugging the reconstruction of the data simulated in divergent pointing mode and thanks to it I found out that the plotting in the `GroundFrame` with the `ArrayDisplay` is working only because so far we have always used simulations with a 20° zenith angle for testing. The bug in the plotting tool in `ctapipe` came out using simulations at 60° of zenith angle and some ad-hoc simulated data at 150° of azimuth and 60° of zenith angle, while it's very difficult to see it with a simulation done at 20° of zenith.

The reconstruction improvement was observed on a event where only two telescopes are surviving the cleaning (see figure 5.18), since the line intersection of just two telescopes is exactly the position of the reconstructed impact point (there's no weighted average with other telescopes).

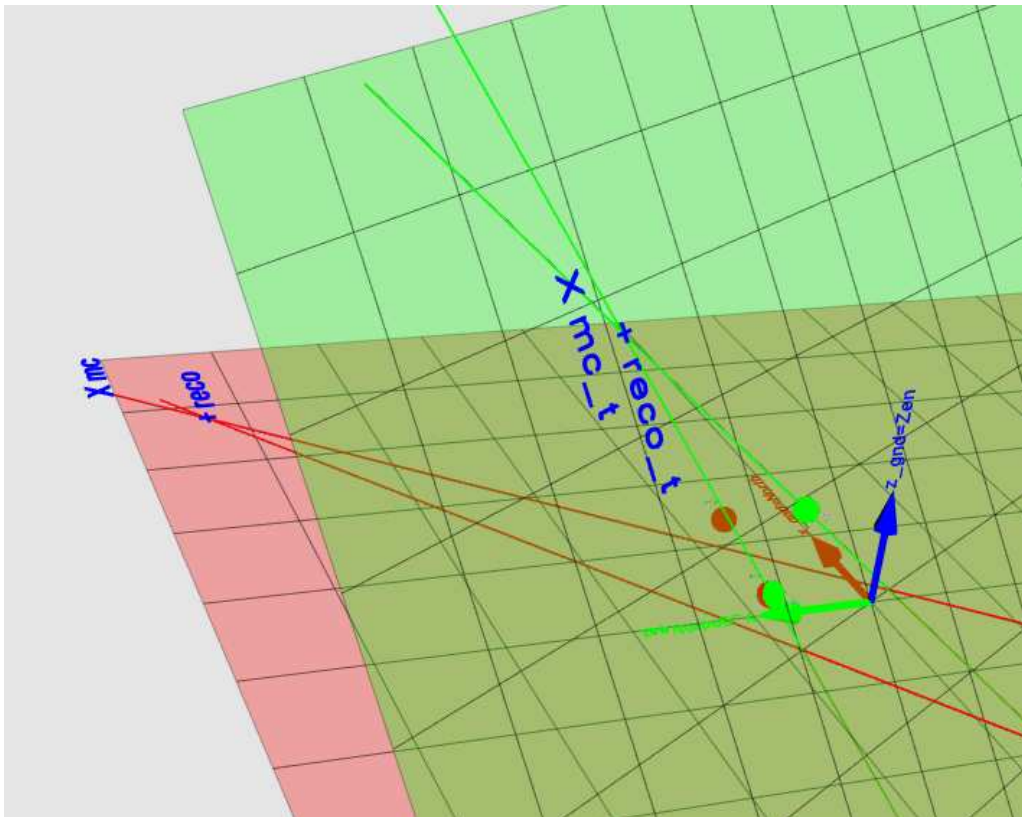


Figure 5.18: image created with the 3D rendered in VTK. The green plane is the **TiltedGroundFrame**, while the red one is the **GroundFrame**, both with the telescopes positions, the lines from the hillas parameters and the impact points from the Monte Carlo simulations (**mc** and **mc_t**) and from the reconstruction (**reco** and **reco_t**). Even though the reconstruction is not perfect, both frames are displaying the lines properly.

Chapter 6

Optimizing for computing

In this chapter I'll focus on the work done on the profiling and the optimizations for the ImPACT pipelines with the goal of improving both the performances and the computing time. The final goal was the porting of the ImPACT routine on a Graphics Processing Unit in order to use such method for the analysis of the LST-1 data when the telescope is observing in mono mode.

6.1 Profiling and optimizations

An important task towards the optimization of the computing time of the ImPACT pipeline, was the execution of the code together with a profiler: this is important since we can clearly see which are the functions that are taking the most of the time and then and, in a second step, we can try to optimize them.

The time profiling was done using both the time profiling functionality embedded in the professional version of PyCharm, which is an Integrated Development Environment (IDE) used in programming and especially for the Python language, and a Python profiler. This profiler *cProfile* is a C extension with a small overhead that makes it suitable for profiling long-running programs. The overhead, the increase in computing time which comes from using the profiler, must be as low as possible in order to have a negligible impact on the time profiling of the pipeline. The first profiler was more useful to get a clear map of the function calls and the number of times those functions have been executed, while the second one was easier and faster to use since it's easy to move in the output which is provided by the profiler. The *cProfile* module can be used in this way from command line:

```
python -m cProfile [-o output_file] myscript.py
```

and the `output_file` which is produced from the profiler can be viewed with SnakeViz (<https://jiffyclub.github.io/snakeviz/>), a browser based graphical viewer for the output of Python's *cProfile* module (see figure 6.1).

This is how the optimization of a pipeline is performed: a benchmark profiling is saved at the beginning, some parts of the code are changed because they might be those slowing

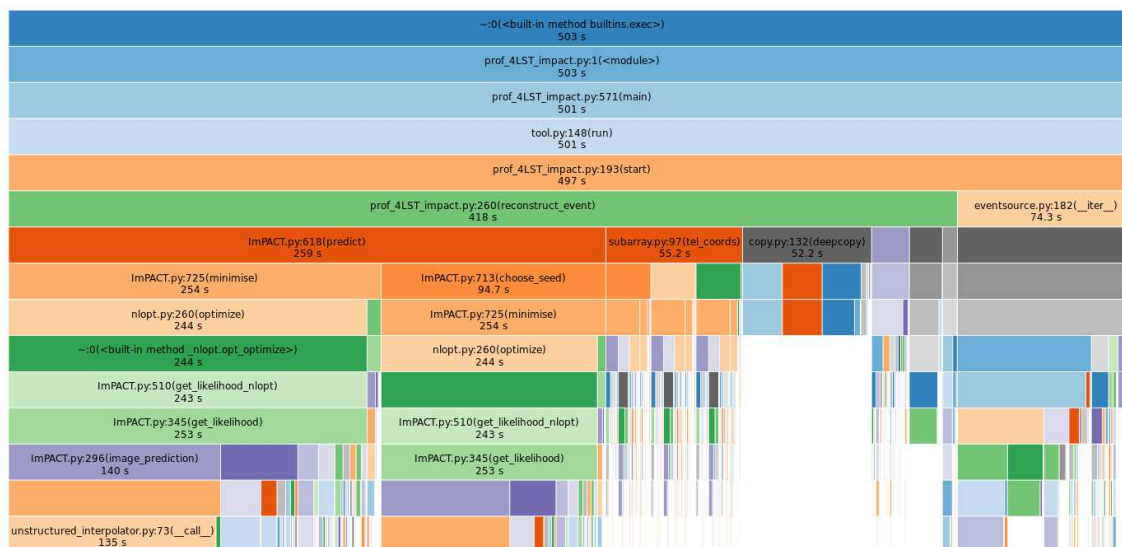


Figure 6.1: snapshot from SnakeViz for a Monte Carlo simulation of gamma-rays observed by the 4 LSTs in Paranal. The total number of events analysed in 235.

the code execution and then the code is re-executed and the profiling is saved. The first and the second benchmarks are then compared to see if the execution time of that specific part of the code that we were tackling has changed.

While optimizing a pipeline it’s important to always look at its output, since the goal of an optimization strategy is to lower the computing time without having a negative impact on the results of the pipeline, which should not change between different runs (or change by a small amount which is not affecting the final conclusions).

In the next paragraph I’ll present the actions taken for the optimization of the pipeline that I received as an example to do the ImPACT analysis with ctapeipe.

Telescope coordinates The pipeline is looping over the events that have triggered at least one of the selected telescopes and each event is treated separately. All the informations that are necessary for analyzing an event are saved in the event itself, but some of them are repeated in each event and not saved in the metadata file. The positions of the telescopes on the ground is fixed and therefore it’s not useful to re-compute it for each event (as it was done at the beginning), but it can be computed just for the first event, stored and re-used for all the other events. The simple removal of this computation from the loop over the events brought made the reconstruction of the event 10% faster. Even though this is a small improvement, it goes in the right direction of reducing the computing time of those computations that are not in ImPACT: this is useful because the percentage of the total computing time dedicated only to ImPACT increases and any further optimizations on the ImPACT code has a bigger influence on the overall computing time.

Furthermore, the coordinates were created using a deprecated method in ctapeipe, such as this,

```

grd_tel = GroundFrame(x=event.inst.subarray.pos_x,
                      y=event.inst.subarray.pos_y,
                      z=event.inst.subarray.pos_z)

```

which was, for each of the three coordinates, extracting the whole set of 3D coordinates and picking just the x, y or z value. An analysis of the implementation of the `ctape.instrument.subarray` module revealed that those `pos_x`, `pos_y` and `pos_z` were deprecated and that where all calling the `tel_coords` function: the code was changed to this,

```

ground_positions = event.inst.subarray.tel_coords
grd_tel = GroundFrame(x=ground_positions.x,
                      y=ground_positions.y,
                      z=ground_positions.z)

```

so that the coordinates were stored in one variable and each of them called from that variable. This proved later on to be not necessary since the `ground_positions` variable was already storing the coordinates in the `GroundFrame` and therefore the creation of the coordinates of the telescopes in the `TiltedGroundFrame` could be done just with

```

# pick place and time to create observation frame
location = EarthLocation.of_site('Roque de los Muchachos')
obstime = Time('2018-11-01T02:00')
horizon_frame = AltAz(location=location, obstime=obstime)

# pointing direction of the array
array_pointing = SkyCoord(az=event.mc.az, alt=event.mc.alt, frame=horizon_frame)
tilted_frame = TiltedGroundFrame(pointing_direction=array_pointing)

# transform coordinates
ground_positions = event.inst.subarray.tel_coords
tilted_ground_positions = ground_positions.transform_to(tilted_frame)

```

at the beginning of the analysis and later on reuse the `tilted_ground_positions` variable.

The computation of these coordinates is mandatory in the ImPACT analyses since the positions of the telescopes (passing the cleaning procedure) in the `TiltedGroundFrame` must be provided for each event: once all the positions are computed it's easy to select those of the telescopes that are being analysed in each event. This computation is not needed when using the `HillasReconstructor` for a standard analysis, since it is done internally in the class itself.

These changes in the computation of the coordinates of the telescopes have decreased the total computing time in the pipeline by 8–10%.

2D to 3D reconstructor The ImPACT analysis needs to start from a seed which is usually the output of a previous reconstructor, such as the `HillasIntersection` (the 2D

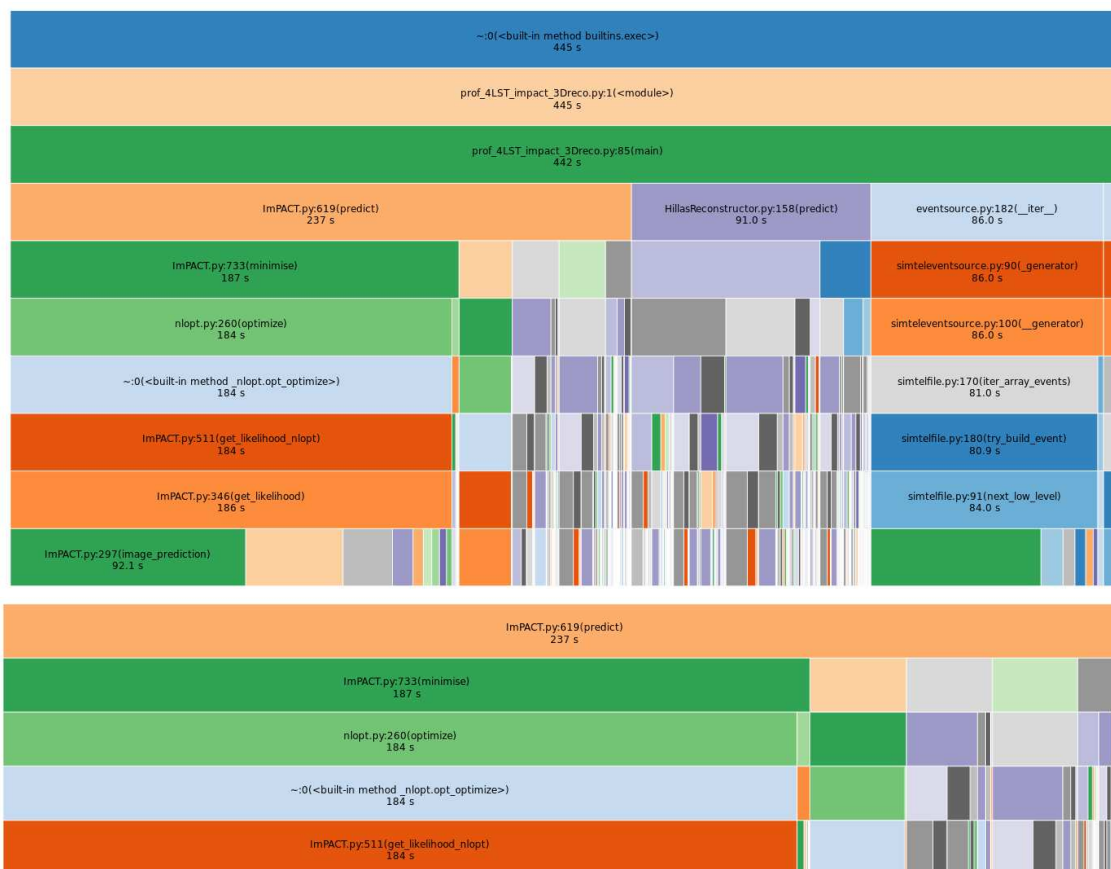


Figure 6.2: snapshots from SnakeViz for a Monte Carlo simulation of gamma-rays observed by the 4 LSTs in Paranal: upper panel is the full profiling while the lower panel is a zoom on the ImPACT section. The total number of events analysed in 235. The ImPACT pipeline at this level is using the output from the `HillasReconstructor`. The I/O operations are also a bit more time consuming at this stage since the input data are not read anymore by the `pyhessio` but by `pyeventio`, which is less memory consuming but a bit slower than `pyhessio` (see details in section 5.1.2).

reconstruction method) which I’ve been using for the first tests with ImPACT. The other method implemented in `ctapipe` is the `HillasReconstructor` which I’ve also adapted and used for the analyses of the divergent pointing data.

Since the 2D reconstruction method was not easy to use (see section 5.2.1) and its performances were expected to give worst results when compared with the 3D reconstruction method, I decided to refactor the ImPACT pipeline in order to use the input from this reconstructor, which resulted in a pipeline which was easier to read with respect to the previous version. The profiling of this new pipeline with the 3D reconstructor, together with a zoom on the ImPACT part can be seen in figures 6.2.

ImPACT optimization After having optimized the ImPACT pipeline I started to work on the optimization of the ImPACT code itself which is calling many classes and methods in `ctapipe`, as described in the previous paragraphs. The first profiling obtained with the 3D reconstructor (see figure 6.2) showed that some of the computing time was spent in the re-creation of the `NominalFrame` twice per event (the first the position of the seed in the

horizon frame is put in the `NominalFrame`, the second time the reconstructed parameters are put in the `NominalFrame` to be projected in the horizon frame), while this frame is not changing for different events in the same file and it can be reused¹. The creation of the frame has been moved to the `__init__` of the `ImPACT` reconstructor and both the array pointing direction and the `NominalFrame` are now class attributes. Moving the initialization of this frame to the `__init__` had no effect on the output of the pipeline.

The second piece of code which was touched by this optimization procedure was the `choose_seed` method inside the `ImPACT` class which was intended to take the seeds given as input to the analysis and to spread them over a larger range of values. Between the different sets of initial seeds, the one with the lower goodness-of-fit was used as the initial seed for the full analysis. This was done only if the number of telescopes in the event (not only triggered, but passing the cleaning and being used in the reconstruction) was 2 or 3, while this procedure was ignored if there were at least 4 or more: the idea behind was that if there are few telescopes, it would be a good idea to first refine the seeds and then to do the complete minimisation procedure.

The idea to remove this part of code in order to use the input seeds directly for the minimisation procedure came from observing that the reconstruction with `ImPACT` was better than 3D reconstruction method only for events with at least 4 telescopes triggered, while it was giving worst result in the case of 2 or 3 telescopes. This piece of code is different with respect to what has been done for the analysis of `ImPACT` for the Mono LST-1. In the mono analysis, a new set of input seeds for the analysis was used to do a new analysis each time, while here the spread over the parameters was happening inside the class before the event was analysed.

The removal of this piece of code brought two results: since that piece of code and the first rough minimisation procedure was not executed, the whole pipeline was a bit faster and the results which was giving as output were better than those obtained in the first benchmark (see figure 6.2).

The first profiling in figure 6.2 can be compared with the new one in figure 6.3 obtained after the two optimizations described above.

The two corrections made the `ImPACT` part 15% faster, with an improvement of less than 10% in the overall analysis.

Bug fixes in `ImPACT` Each event must be treated separately and must not carry informations on the previous event, otherwise the analysis might end up being biased in some non-predictable way.

While debugging the `ImPACT` class, the `TemplateNetworkInterpolator` class and the `UnstructuredInterpolator` class I found out that some variables which were initialized equal to 0 at in the `__init__` of the `UnstructuredInterpolator`, they were being cor-

¹This is true for MC events, where the pointing direction of the telescopes is not changing over time. In a real scenario, the telescopes will be tracking the source. Even in this case, it would only be necessary to update the frame attributes without creating a new frame for each event.

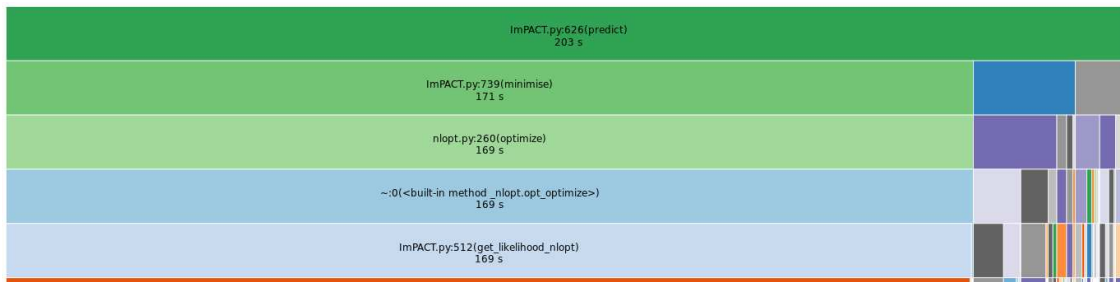


Figure 6.3: snapshot from SnakeViz for a Monte Carlo simulation of gamma-rays observed by the 4 LSTs in Paranal. This snapshot describes only the ImPACT part.

rectly overwritten in the steps of the minimisation procedure since at each new iteration of the minimisation algorithm there’s a new call to the **UnstructuredInterpolator** and those values are updated. The analysis of the first event was done properly but, starting from the second event, the values which were supposed to be re-initialized equal to zero for every new event were instead keeping the last value from the previous run.

In order to solve this bug, a **reset** method was created in the **ImPACT** class and called inside the **predict** method of the **ImPACT** class which is called at the beginning of each event. This reset method is calling a **reset** method in the **TemplateNetworkInterpolator** which is calling the **reset** of the **UnstructuredInterpolator** in which those variables are set equal to zero.

The variables which are set to zero are **self._previous_v**, **self._previous_m** and **self._previous_shape** which are used to move in the parameter space with the methods and parameters provided with the Delauney algorithm and therefore it’s important the reset them at the beginning of each new event (see section 5.2.2). The fix applied now ensures that the fitting between the different events are now completely independent, which was not the case before the fix.

6.2 ImPACT on GPU

Due to the long execution time for the ImPACT pipeline implemented for the analysis of the LST-1 data, I decided to reimplement the ImPACT analysis in order to use the computing power of the Graphics Processing Unit (GPU). A GPU is a specialized electronic which is more efficient than a CPU in doing highly parallel calculations: GPUs are responsible for creating the output to be displayed on the screen of a computer but they can be used not only for gaming or rendering application but also for pure scientific computations.

While CPUs can have up to few tens of cores, a GPU can easily have thousands of cores, which means that if there is a computation which has to be done on many independent items of a dataset, each item can be computed by one of the cores of the GPU and the result collected after each core has finished. An example of a highly parallel calculation can be for example the calculation of the square of an array of values, where each element of the array is independent and therefore each calculation can be done by a different core.

One of the main bottlenecks that can raise when using a GPU is in the memory transfer between the RAM and the memory of the GPU, which is the place where the cores of the GPU will be looking for the data to use in the computation. Once some data has been transferred into the memory of the GPU they shouldn't be transferred back and forth from the the GPU's memory to the RAM, but they should stay in the GPU's memory and transferred back to the RAM once the computation has been completed or it really necessary to transfer it back.

I decided to port ImPACT on GPU both because the computation are rather simple and are usually repeated on many different templates over the whole analysis. Due to the missing informations from a reconstructor in the mono analysis for the LST-1, the first idea that came in mind was to remove the minimization procedure in which only one template is being used in each step of the minimization procedure. The idea of using the GPU would be to try all the templates at once and then using the output from the goodness-of-fit as a weight to get the final estimates on the reconstructed parameters.

6.2.1 PyTorch implementation

Since the ctapipe library is written in Python, I decided to refactor the ImPACT class using the same programming language and a very powerful Python library called PyTorch (<https://pytorch.org/>). PyTorch is an optimized open source machine learning framework for deep learning using GPUs and CPUs based on the Torch library and mainly developed by Facebook AI research group. PyTorch is being used in different fields (also the GammaLearn framework [99] is developed with PyTorch) especially for Deep Learning applications, while one of the most important high level features of the framework is the tensor computing (like Numpy) with a strong acceleration via GPU. Similar to the `n-dimensional ndarray` in Numpy, the tensors in PyTorch are multi-dimensional matrices containing elements of a single data type. These structures are very useful in a Python pipeline because a lot of operations that would need looping on the single elements of the arrays can be done in a faster way on the whole Numpy `ndarray`, or even faster on a GPU with the PyTorch tensors.

Due to this strong similarity between Numpy and Pytorch it's relatively easy to port single functions from the first one to the second one and to speed-up the code, whereas a deeper optimization of complex pipeline might need a complete refactoring to get more out of the computing capabilities of a GPU.

It's important to notice that PyTorch is developed to be working only with GPUs made by the NVIDIA company, that can be programmed using the CUDA platform and API model created by the same company. The computer used in these analysis is equipped with one NVIDIA GeForce 1050, with 640 CUDA cores and 4GB of dedicated memory: the computer has a i7-7700HQ CPU with 4 cores.

Template loading The first thing to do is to load the templates on the memory of the GPU, so that once they've been loaded at the beginning of the analysis, they can just be used in each event. A problem with the template loader into the GPU memory can raise if the templates are not loaded properly, which can result in a high memory usage. In the first rough implementation of the template loading routine, the template were loaded one at a time by means of a loop over the whole dataset (this implementation was done using the `torch.utils.data.dataset.Dataset` class). With this implementation almost 100% of the GPU memory was being used, probably because the templates were scattered on the memory and not loaded contiguously.

The final implementation was done using both the `dataset.Dataset` class and also the `dataloader.DataLoader` from `torch.utils.data`, the latter being the core of PyTorch data loading utility, with a lot of important features especially for Deep Learning. The templates for the LST camera are loaded in RAM using `gzip` and `pickle` in the same way it is done in the standard IMPACT class in `ctapipe` and then a subclass of the abstract class `dataset.Dataset` is created (called `ImpactDataset` class), which inherits the methods and properties from the parent class: the methods `__len__` and `__getitem__` from the `Dataset` class must be overloaded when defining the own dataset class. The templates are arranged in a dictionary structure, using the parameters for each template as the *key* (called “label” in the class) to retrieve the corresponding template (called “image”). This is the class implemented for loading the templates:

```
class ImpactDataset(Dataset):
    def __init__(self, path):
        dict_in = OrderedDict(pickle.load(gzip.open(path)))
        self.keys = torch.tensor(np.array([k for k in dict_in.keys()]))
        self.values = torch.tensor(np.array([v for v in dict_in.values()]))
    def __len__(self):
        return len(self.keys)
    def __getitem__(self, id):
        return {'image': self.values[id], 'label': self.keys[id]}
```

```
template_file = "path_template/LST_05deg.template.gz"
dataset = ImpactDataset(template_file)
```

An instance of this class is then passed to an instance of the `dataset.DataLoader` class, which takes into account all the parameters of the data loading, and then the templates can be loaded into the GPU memory:

```
device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
dataloader = DataLoader(dataset, batch_size=len(dataset),
                        num_workers=3, drop_last=True)
for batch in dataloader:
    images = batch['image'].to(device)
```

Loading the templates takes roughly 2 seconds and 70% of the GPU memory (2.8 GB).

The parameter `device` is fundamental in any PyTorch pipeline because it is responsible of moving data to the GPU and retrieve data from it: if the same software is used on a machine with no GPU support, then the CPU will be used. If the GPU is used, then the `images` object is in the GPU memory and any calculation done using these data, will be executed on the GPU.

Preliminary tests I did some simple preliminary tests in order to understand the differences in performance between a computation done on the GPU compared to the same computation executed on the CPU with PyTorch, and compared this with a Numpy implementation. This test has been executed on all the templates, after loading them in memory (either in RAM or on the memory of the GPU), and the simple computation which has been executed is the average of each template.

In the Numpy implementation, without using any PyTorch function, a loop over the templates (with the computation of the average of the template) took $1.01 \text{ s} \pm 3 \text{ ms}$ to end. The same calculation done using the `ImpactDataset` class developed using PyTorch and in which the calculation has been executed on the CPU, used a total of $115 \text{ ms} \pm 4 \text{ ms}$. The time needed from the computer to perform this calculations, which is the time that I'm reporting here, is also called "wall time". The "CPU time" instead takes into account both the amount of time the CPU spends performing some action for a program and the time the CPU spends performing system calls for the kernel on the program's behalf. Comparing the "wall time" and the "CPU time", we can see that while in the Numpy implementation these number are almost equal, in the PyTorch implementation the CPU time is 4 times greater with respect to the wall time. This happens because PyTorch can spread the calculation of the 4 cores of the CPU that I'm using for these tests, while a simple Numpy implementation is using only one core of the CPU. From these simple tests we can see the efficient data handling of PyTorch is giving a speed-up of 2 to the computing time (from one second to roughly 500 ms) and the 4 cores in the CPU are giving another speed-up of 4 times to the overall calculation, resulting in a total speed-up of 8 times, without even using the GPU. This speed-up, which has been easily achieved using only some PyTorch functions, could in principle be achieved with more effort dividing the dataset into 4 parts and using one core per each part, but this implementation was not tested in this work.

The test of this simple calculation using PyTorch to perform this calculation on the GPU has been done with this piece of code:

```
start = torch.cuda.Event(enable_timing=True)
end = torch.cuda.Event(enable_timing=True)

# begin recording
start.record()

# average for each template
```

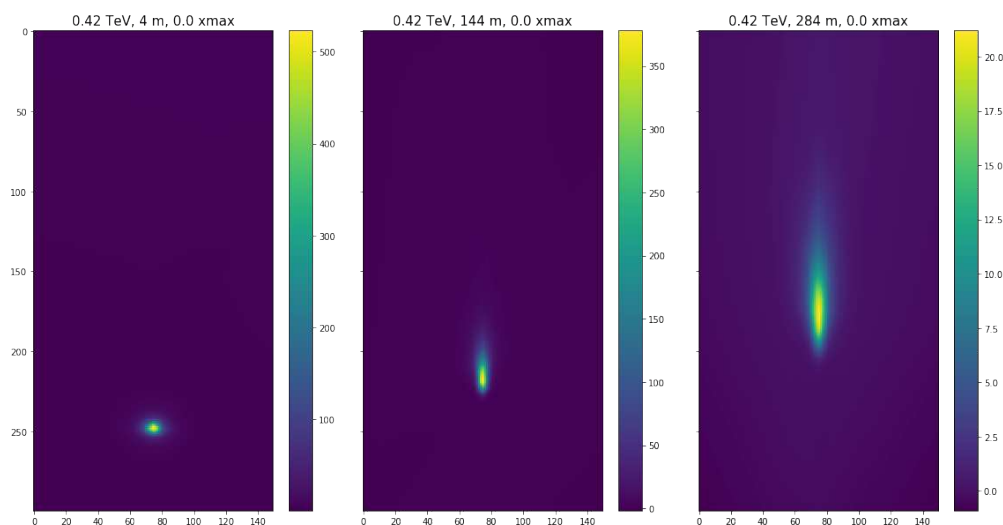


Figure 6.4: 3 example templates for the LST camera, with a size of 150x300 pixels.

```
torch.mean(images, dim=(1,2))

# record end and synchronize
end.record()
torch.cuda.synchronize()
gpu_time = start.elapsed_time(end)
print(f'gpu_time = {gpu_time:.2f} ms')
```

where `images` are the templates and the `torch.cuda` module is a wrapper for using the CUDA functionalities in this Python program. The `Event` method, together with the `torch.cuda.synchronize()` function, is used for timing the execution time of a computation on GPU. Since the computation is spread over the various cores of the GPU and not all of them finish the computation at the same time, it's important to use wait for all the computations on the GPU to finish (using `synchronize`) and record the time only after this synchronization is finished ². The computation on the GPU takes roughly 25.1 ± 0.2 ms, bringing a speed-up of 5 with respect to the implementation on CPU with PyTorch and it is almost 40 times faster with respect of the Numpy implementation.

The implementation The templates have a size of 300×150 pixels (see examples in figure 6.4), with a pixel size of 0.02° , meaning that the size of the templates is $3^\circ \times 6^\circ$. The first thing to do is to assign meaningful coordinates to the pixels in the template in order to be able to compare them to the camera image. The origin of the camera lies in the pixel with coordinates (75, 250), so that the x-axis extends from -1.5° to 1.5° , and the y-axis ranges from -5° to 1° .

In order to create an image from the template to be compared with the camera image, the camera image must be properly rotate and translated according to the Hillas

²The PyTorch implementation seems to invert the synchronization and the end time, but this is how a correct time measurement has to be coded in a PyTorch following the documentation.

parametrization. Thanks to this rotation and translation of the pixels the pixels from the template and the camera image will be overlapping. The pixels in the camera passing the cleaning are superimposed with the template, which has a pixel size which is much smaller with respect to the pixel size of the camera, resulting in many pixels from the image template falling under a camera pixel.

The idea is to compute the distance between each pixel in the template from each pixel in the camera in order to find for each pixel in the template which is the closest camera pixel. Since each pixel in the template knows which is its closest camera pixels, but many template pixels will fall under the same camera pixel, the signal estimated for each pixel in an average of the underlying template pixels (see figure 6.5).

I came up with many different implementation to this algorithm creating the image, the fastest using roughly 3 ms per each template (depending on the number of pixels passing the cleaning) to create the template and compute the likelihood of the template being the best representing the camera image. The templates are more than 7000, meaning that the total computing time is about 21 second, which is way more that the 10 seconds needed in the standard routine. The creation of the prediction from the template is roughly taking the same time as the time needed in the standard ImPACT implementation on CPU, but the GPU implementation is being applied to all the templates, while the standard version is using only some of the templates (few hundreds at maximum, depending on the minimizer which is exploring the parameter space of the template at with a new trial for each iteration of the minimization process), resulting in a total computing time of 1 second to find the estimate of the shower parameters.

There can be many motivations for which the GPU implementation hasn't brought a speed-up. First of all, the handling of the memory and of the computations to be done on the GPU, is done entirely internally in PyTorch and the level of detail that can be achieved using the CUDA extension in the C programming language cannot be easily achieved in PyTorch. For example, even a simple computation such as the product between two matrices, might be achieved with different computing times according to the fine details in the handling of the computing kernel to be used in the computation on the GPU. A second motivation can be the non optimal implementation of the algorithm explained here, so that a further optimization can be achieved with a deeper time profiling and a re-implementation of the algorithm used here. The standard iterative process, implemented in `ctapipe`, has not been exported on GPU because I thought that this implementation in which the new step depends on the previous one would not use the computing power of the GPU and would only slow down the computation: this could also be a possible speed-up for the analysis.

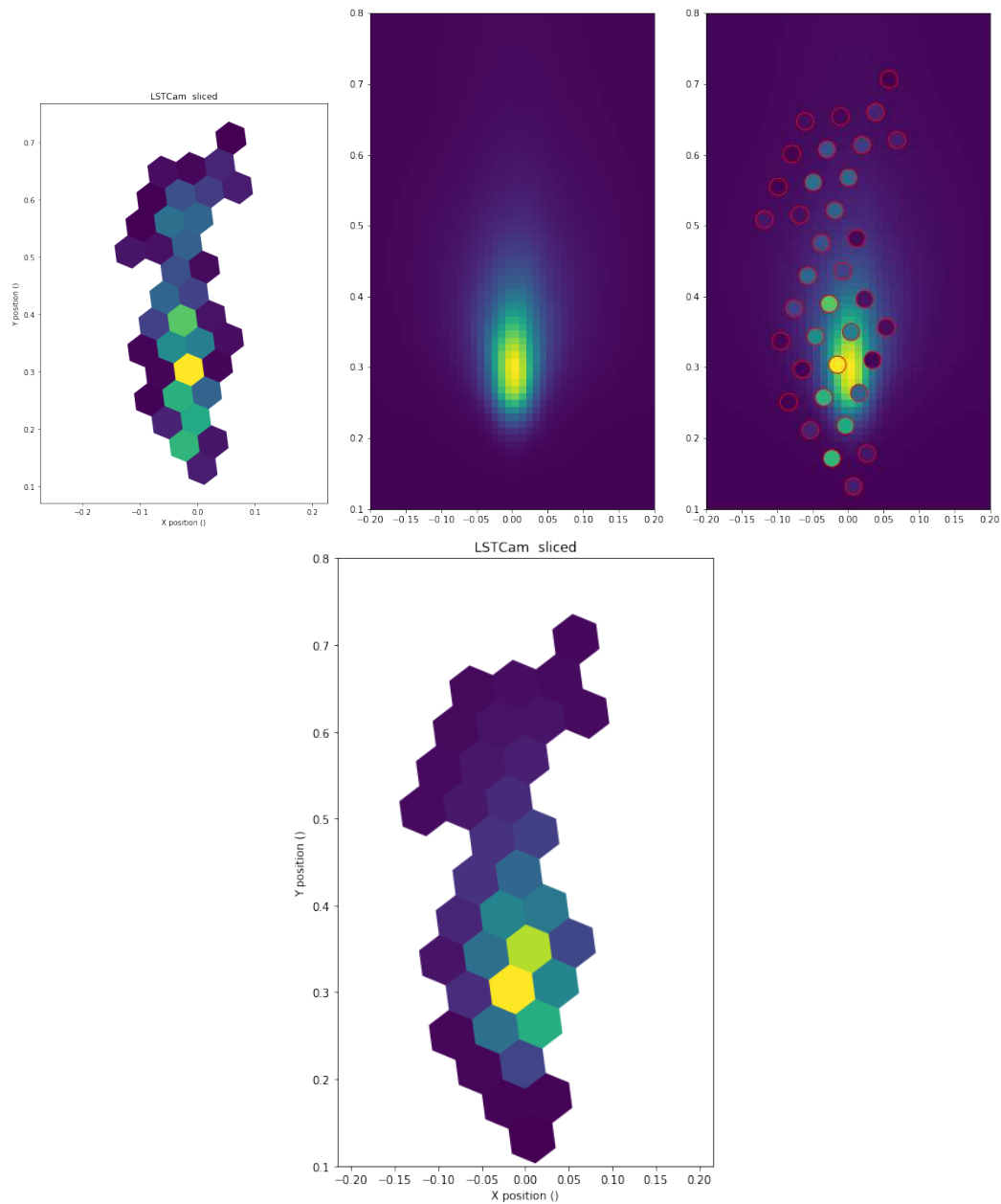


Figure 6.5: The creation of the predicted image using the template and the positions of the pixels passing the cleaning. *Upper panel:* The left image represents the pixels passing the cleaning, that have been properly rotated according to the rotation angle obtained in the Hillas parametrization. On its right there is a zoom of the selected template (the one matching the most with the camera image). The third image is a superposition of the template pixels and the camera pixels. *Lower panel:* The image in the bottom panel is the prediction obtained combining the camera pixel positions with the values in the template.

Chapter 7

Optimizing for Physics

In this chapter I'll be presenting the work that I've been doing on the estimation of the performances of CTA under a new observing mode called "divergent pointing". CTA will be the first IACT array able to use this pointing mode due to the large number of telescope that will be built in the northern and the southern sites with respect to the existing instruments. I'll briefly introduce different pointing methods and then I'll focus on the improvements that an observation done in divergent mode can have for some science goals, such as the surveys and follow-up of gravitational waves events. Then I'll present the code that I implemented in `ctape` in order to use this software package to analyse the Monte Carlo simulations in which the telescopes are pointing in divergent mode. Finally I'll present the preliminary results that I've obtained for different divergent pointing configurations. The work was carried on together with Alice Donini, PhD student at the University of Udine, and other colleagues part of the Analysis and Simulations Working Group of CTA.

7.1 Pointing modes and divergent

In the standard pointing scheme, that has been used so far in all the existing IACT experiments, the observations are performed with all the telescopes pointing to the same direction in the sky, which means that they observe the same portion of the sky. This pointing mode will be referred to as "standard" or "parallel" since the axis of the telescopes are non intersecting. In this pointing mode, the field of view of an array with the same telescope's types is the same as the one of a single telescope and the various fields of view of the telescopes in the array are almost completely overlapping. If some bright stars happen to fall in the field of view of the telescopes that are doing the observations, those stars will fall in the same positions in the cameras (provided that the telescopes have the same focal length).

The total field of view of an array of telescopes can be significantly enlarged by slightly varying the pointing direction of the telescopes according to some pointing pattern. The performances of the array can vary a lot according to the pointing pattern which is chosen

and the study of the performances of the array need to be assessed with Monte Carlo simulation.

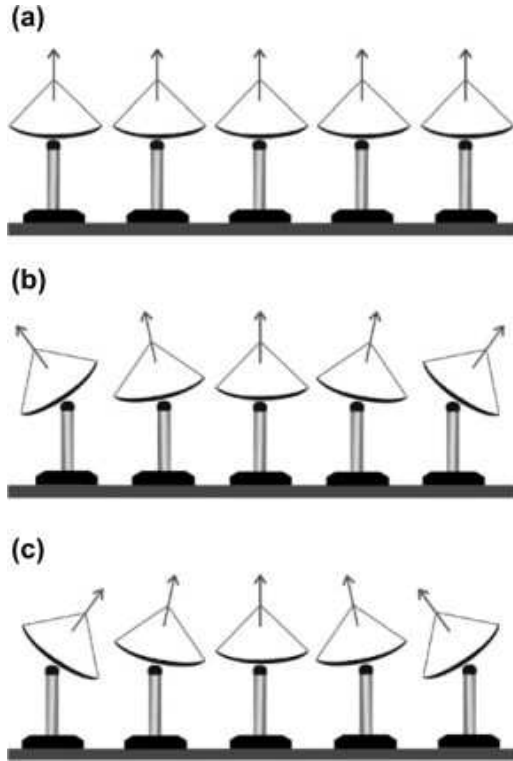


Figure 7.1: Standard pointing (a), divergent pointing (b) and convergent pointing (c). From [100].

There is a trade-off between the total field of view of the array and the average telescope multiplicity. The telescope multiplicity is the number of telescopes that are used for the reconstruction of a certain event and it is a crucial parameter for the quality of the shower reconstruction, since the more telescopes are observing an event the better is the reconstruction of the parameters.

Since the chosen telescopes' pointing will affect the angular resolution, the energy resolution and the sensitivity, these performances curves have to be studied in order to optimise the pointing pattern of the divergent configuration. The science case will fix the requirements for energy and angular resolution and sensitivity and the corresponding pointing scheme will be chosen accordingly.

The parameter space is very large, due to the number of telescopes and the possible pointing modes that can be implemented and studied. In this work we tried a simple implementation of a “divergent pointing” mode in which the telescopes are pointing in the outward direction by an angle increasing with the telescope distance from the array center. This ensures that the pointing axis are not intersecting at a certain height above the ground level. The pointing mode in which the axis of the telescopes are intersecting is called “convergent pointing”, it has been studied in [100] and it will be presented in the next paragraphs.

Other pointing modes, such as pair or triplets of telescopes having the same pointing directions (a mix of parallel and divergent pointing) across the hyper field of view of the array are also possible, but they haven't been investigated in this thesis.

The pointing direction must not be confused with the focusing of the telescopes, usually placed at around 12 km above sea level, which is the height at which a shower with an average energy reaches the maximum of the emission. The focus of the H.E.S.S. II camera can be adjusted moving the camera along the optical axis of the telescope and it can have a huge impact on the data acquisition and analysis [101].

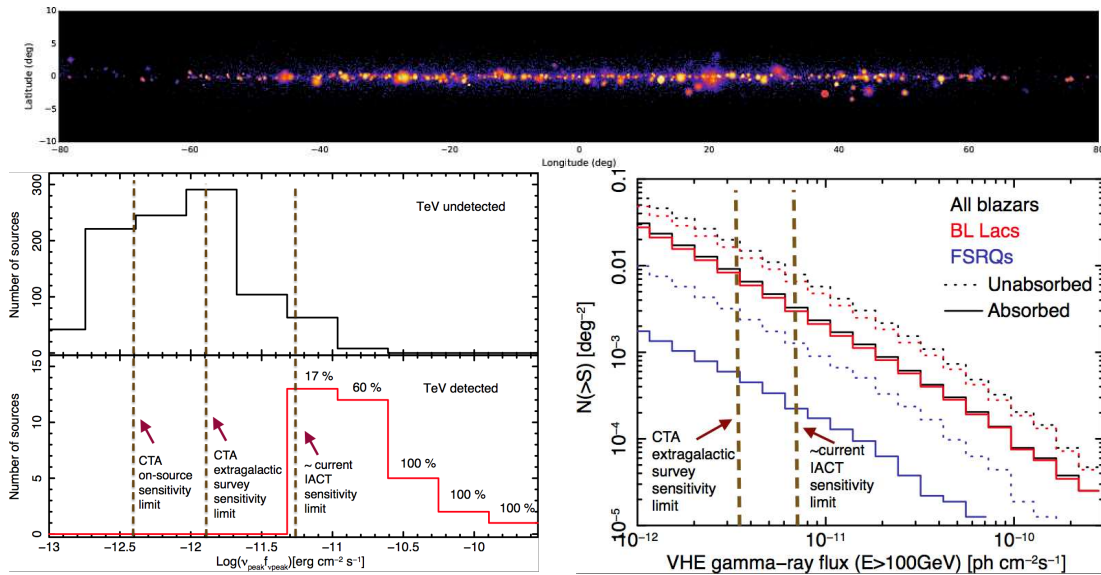


Figure 7.2: *Upper panel:* inner region between -80° and $+80^\circ$ in Galactic longitude as it will be seen by CTA adopting the proposed GSP KSP observation strategy and using a source model which incorporates supernova remnants, pulsar wind nebulae and a diffuse emission. *Lower panel:* prediction of the number of blazars on the sky in the GeV – TeV domain as a function of the peak synchrotron (left) and integral gamma-ray flux above 100 GeV (from figures 1.2 and 1.3 of 2).

7.1.1 Goals

There are many science goals that would benefit from using the divergent pointing mode instead of the standard one, the surveys of the Galactic plane and of the extragalactic sky being some of them [102].

Due to the immediate effect of the divergent mode of increasing the total field of view of the array (a hyper field of view, HFoV), the serendipitous detection of a transient source happening in the field of view while the array is performing a survey may be more likely to happen due to a much larger field of view available. The drawback of the divergent pointing is the worsening in the sensitivity, but if a transient source which happens in the HFoV is bright enough, a bad sensitivity is good enough to have a detection. Shorter pointings with a deeper sensitivity would allow to see fainter sources, but the chances would be much smaller with respect to the divergent pointing.

Another important science project in which the divergent pointing would play an important role is the electromagnetic follow-up of gravitational waves events, since those events produce probability sky maps that are usually several times larger with respect to the field of view of IACT instruments [24].

Galactic and extragalactic surveys

We can split the types of sources that will be observed with CTA into three groups: those sources that have already been observed at VHE or that we know they can produce VHE radiation, the alerts received and the blind survey mode. The idea of performing a survey

at VHE is to observe a certain patch of the sky looking for sources in a blind way in order to provide an unbiased view of the Universe at energies above tens of GeVs, without relying on external triggers from other experiment and satellites, eventually being able also to send alerts. The area of the sky that the surveys are going to cover is much bigger than the field of view of an array pointing in standard mode, a huge number of pointings are needed and their durations have to be properly scheduled in order to achieve a certain “survey depth” in all the sky region, i.e. the minimum source flux detectable. The pointings will even more than those required to simply cover the survey area because in order to cover this area with a equal response over the whole region of the sky, it’s necessary to overlap them, resulting in a greater number of pointings.

Due to the limited duty cycle of IACT experiments and the several science projects that have to be covered, some studies have been performed in the past in order to allocate a certain amount of observation time to each of them according to the different requirements.

In order to reach a sensitivity similar to the decade long H.E.S.S. programme on the Galactic plane, i.e. 6 mCrab, over 1/4th of the sky, the amount of time dedicated to the extragalactic sky survey has been set to 1000 hours [24]. No extragalactic survey has ever been performed before using IACTs while ground-level particle detectors, such as HAWC, are naturally being used as survey instruments due to their large field of view (HAWC covers 15% of the sky) even though they have a modest sensitivity, a limited angular and energy resolution. The divergent mode will be used for this KSP if it can be shown that the required integral flux limit can be achieved in the same, or even with a lower observation time (the results in [100] are promising, showing that a convergent mode can also give a better angular resolution).

Another important survey program is the Galactic Plane Survey (GPS) in which the inner galaxy and the Cygnus region will be covered with a sensitivity of ~ 2 mCrab at each point while the entire Galactic plane will be scanned with a sensitivity of ~ 4 mCrab. This survey will be done using a parallel pointing scheme since the lower energy, the angular resolution and the sensitivity in divergent pointing would make it harder to perform a survey with a good sensitivity and to tackle the source confusion problem in the Galactic plane efficiently [24]: this survey will be a KSP for CTA South. In this case the pointing mode might be even different, diverging only along the galactic latitude and not along the galactic longitude, even though a gain in observing time compared to successive standard pointings may not be as great for such unidimensional survey [102].

Serendipitous discovery

If the divergent mode is proven to guarantee the required performances during the extragalactic sky survey, the divergent pointing mode will be a possible observing mode. The divergent pointing would be used in order to increase the chances for a serendipitous detection of a transient source across the larger hyper field of view of the array [24].

The interplay between several parameters must be studied in order to understand if

the divergent mode is a better operating mode. The average telescope multiplicity and its uniformity across the hyper field of view affect the total duration of the observation and the minimum flux that a source must have in order to be detectable in divergent mode.

Several science goals could benefit from carrying the extragalactic survey in divergent mode, such as the observation of the prompt phase of GRBs, the search for VHE transients and the multi-wavelength and multi-messenger electromagnetic follow-ups of gravitational waves events [24].

Gravitational waves follow-up

The 11 GW events observed during the first two observation runs of the LIGO and Virgo interferometers, have revealed a very broad range for the sky localization: the event GW170817, the first observation of a binary neutron star merger, was localized within an area of 16 squared degrees, while the event with the worst sky localization, GW170823, had a sky localization of 1651 square degrees (see figures and table 2.16 from [64]).

When the interferometers detect a gravitational wave, within a couple of minutes an alert is sent to all the observatories and satellites in the multi-messenger network (like the Astrophysical Multimessenger Observatory Network, *AMON*) in order to start the follow-up in all the other wavelengths. CTA will receive the alerts in order to perform the electromagnetic follow-ups at VHE of the GW events, as the other IACT systems have done in the past few years.

The goal of an electromagnetic follow-up is to search for some emission inside the region of sky provided by other experiments, and since CTA telescopes have a field of view of 50 square degrees at most (for MSTs with a field of view of about 8 degrees), those events with a sky localization which is larger than this value have to be followed according to some pointing pattern with a certain time spent per each pointing plus some technical time needed for data acquisition and slewing of the telescopes. Those pointings and their durations have to be determined in order to maximise the probability to find the source inside the selected sky region. The probability sky map provided by the interferometers have to be convolved with the 3D distribution of galaxies (inside which a GW event will happen) inside that region of sky in order to take into account the distance of the possible source and corresponding flux which would be observed during the electromagnetic follow-up.

If the follow-up is performed in a divergent mode instead of the parallel mode, the big field of view would reduce the number of pointings needed to cover the region of sky. The duration of each pointing has to be optimized and is a function of many parameters, such as the 3D distribution of galaxies, the average zenith angle of the observation and many other. Even though the sensitivity might be worst in divergent pointing, a real-time analysis in divergent pointing would guarantee the detection of the source in some region of the hyper field of view, so that after a preliminary phase in divergent mode, the array can then be repointed and used in a standard pointing mode, with a higher sensitivity, angular

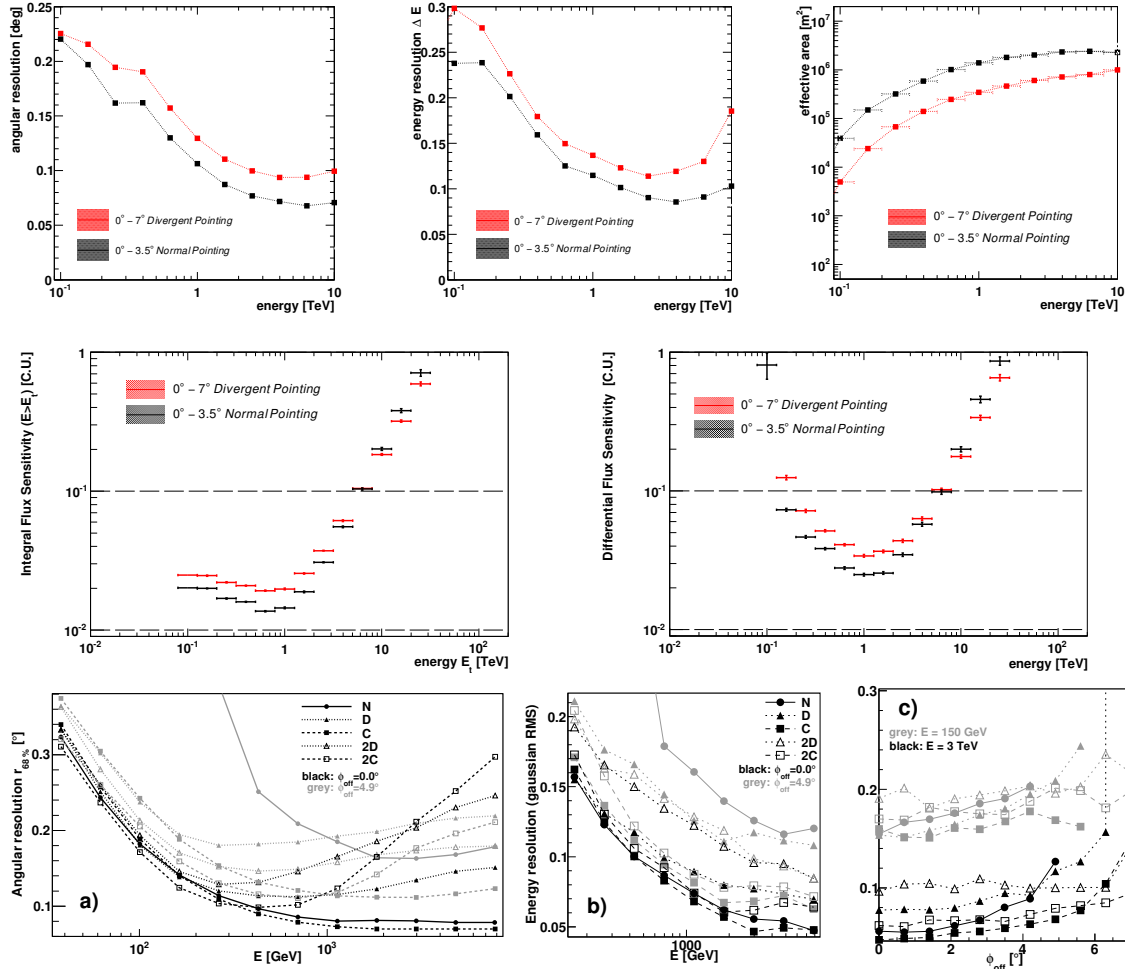


Figure 7.3: **Upper panel** (figure 6 from [103]): the three plots compare a divergent mode, uniform between 0 and 7 degrees from the center of the total field of view, and a parallel mode, which has a uniform response between 0 and 3.5 degrees. The observing time in divergent pointing is 8 hours, whereas the one for the parallel pointing configuration is 2 hours, which is the ratio between the total field of view for the two pointing modes. **Central panel** (figure 7 from [103]): the two plots shows the differential and the integral flux sensitivity in crab units (C.U.). The divergent pointing is 20–25% worse than the parallel mode in the core range, but performs better above 5 TeV. **Lower panel** (figure 15 from [100]): the **N** configuration is the parallel mode, **D** is the divergent one, **2D** has a double divergence angle with respect to **D**, **C** is convergent pointing and **2C** is twice more convergent than **C**. a) is the angular resolution after background suppression cuts, b) is the energy resolution on axis (black) and at 4.9° (grey) for the 5 configurations and c) is the angular resolution at 150 GeV (grey) and at 3 TeV (black) as a function of the divergent angle.

and energy resolution. More details on the follow-up of GW events by CTA can be found in [84] and in section 4.3 for the work done on this topic during the thesis: the estimation of the benefit in using the divergent pointing with respect to the standard pointing in the follow-up of GW events needs further analyses.

7.1.2 Previous works

The work that I will present in this thesis has already been presented in some other works that have been published in the past years.

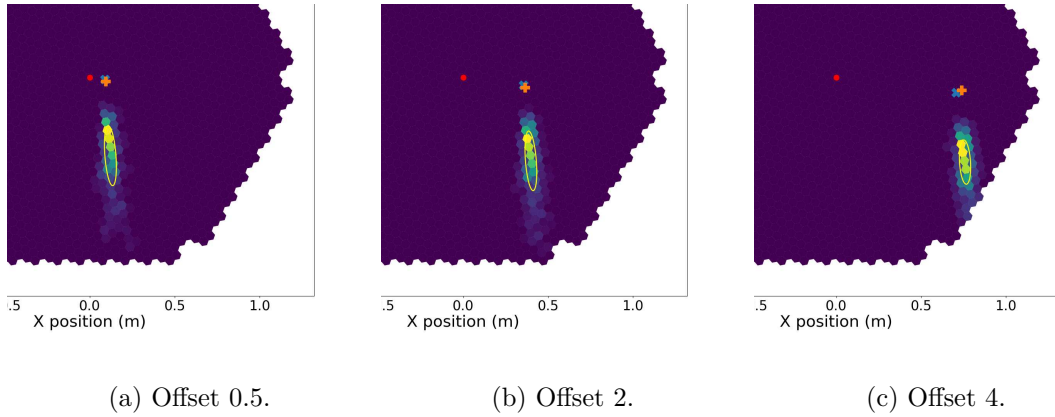


Figure 7.4: The same event seen by the same MST telescope but with different diverging angles (the red dot is the **center of the camera**). Here I'm also plotting on the camera the projection of the sky position of simulated particle and it's reconstructed value (alt-az to camera frame transformation): the cyan cross is the **simulated position** and the orange cross is the **reconstructed position**.

A work performed with an array of 23 MSTs simulated in the H.E.S.S. site in Namibia (see [100]), has presented both a parallel pointing, a divergent mode and a convergent one, going even deeper into the analyses and showing the performances in divergent mode for the sky survey mode (see lower panel in figure 7.3). In case of the divergent pointing, it was shown that the time needed to perform a scan at a given sensitivity is smaller by a factor of 2 with respect to the parallel mode, with the direction and energy reconstruction accuracies for the divergent mode being a factor two worse than other observing modes. Furthermore, it was shown that a convergent configuration is even capable of a better angular resolution with respect to the parallel pointing configuration (see figure 7.3).

In [103], L.Gerard used an array of 56 7m-class SSTs and 18 MSTs using the first MC production and a simple telescope array, comparing a observation performed in normal pointing with one in divergent pointing with a final field of view of about 20° . This work has shown that the sensitivity for parallel pointing is rather homogeneous up to 3.5° , while the sensitivity for the divergent pointing configuration is homogeneous up to 7° (upper panel in figure 7.3): for an observation time of 2 hours for parallel mode and 8 hours in divergent mode, the angular and energy resolution and the sensitivity are 20% to 25% worse in divergent mode than in parallel mode in the core energy range, while the divergent performs even better above 5 TeV (central panel in figure 7.3)¹.

7.1.3 Simulations and analyses

Both works presented above were performed using an old MC production (namely the **prod1**), where the number of telescopes, their type and the construction site was not yet specified; our goal is to continue this work using updated MC models for the telescopes and definitive sites.

¹This result for the divergent pointing is quite strange since angular resolution, energy resolution and effective area are worse than the standard pointing case, it's quite strange to have this behaviour in the sensitivity.

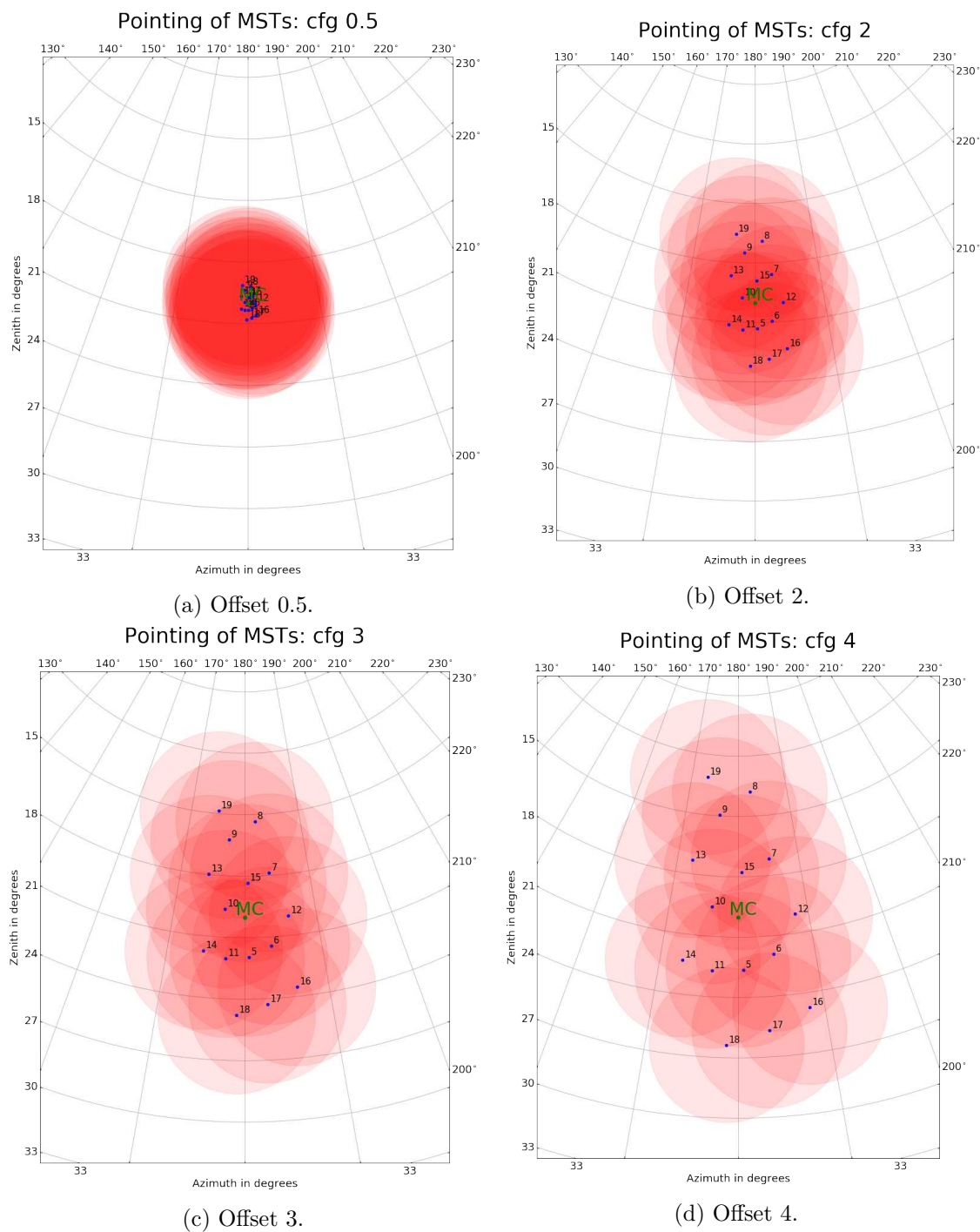


Figure 7.5: Fields of view in polar coordinates for the different “on-axis” configurations that have been tested. The camera used for the MSTs is the NectarCam: the IDs of the MSTs range from 5 to 19. The “offset” is just a parameter used in our script to generate those pointings, and not the actual separation in degrees between the telescopes. The incoming direction of the simulated gamma-rays is 180 deg of zenith and 20 deg of azimuth.

Keeping in mind the results that have been obtained in [100] and [103], we started our work on the divergent pointing creating new configuration files for `sim_telarray` software, using all the parameters from the standard `prod3` MC production a part from the pointing direction for the telescopes.

The pointing of each telescope is specified in the input card, which looks something like this (here I just put the MST number 5 and 6):

```
...
# elif TELESCOPE == 5
#     include <CTA-ULTRA6-MST-NectarCam.cfg>
#     TELESCOPE_THETA=20.33
#     TELESCOPE_PHI=180.08

# elif TELESCOPE == 6
#     include <CTA-ULTRA6-MST-NectarCam.cfg>
#     TELESCOPE_THETA=20.24
#     TELESCOPE_PHI=180.58
...
```

where the `TELESCOPE_THETA` angle is the angle with respect to the zenith direction, `TELESCOPE_PHI` is the azimuth angle (0° means pointing towards North) and the configuration file for the MST with the NectarCam is `CTA-ULTRA6-MST-NectarCam.cfg`.

These are usually common between all the telescopes and are specified just once, while in divergent mode each telescope has a different pointing, which was calculated with another script. In the chosen implementation of a divergent pointing mode we choose to consider the axis of the telescopes as non intersecting, in order not to have a convergent pointing situation between any pair of telescopes. The pointing pattern was produced with a script which is using the positions of the telescopes on the ground to define a divergent pointing pattern. This script was produced in order to have some test configurations to play with and a better script which can handle the telescope multiplicity and the dimension of the hyper field of view of the system is being created in the divergent pointing working group.

In order to test the reconstruction for a divergent pointing configuration, we started producing a simple `sim_telarray` configuration files with a small diverging angle between the telescopes (see figure 7.5a). We installed and properly used both `CORSIKA` and `sim_telarray` in the INFN farm in Trieste and used the configuration files that we have produced to create some simple Monte Carlo simulations. The output from `CORSIKA` is usually piped into `sim_telarray`, but we decided to save it in order to re-use exactly the same simulation with different configuration files, changing the pointings between each of them in order to see how the increasing diverging angles affects the image in the camera (see figure 7.4). Here we focused only on point source gamma-ray simulations, leaving aside other particles simulations, such as protons, electrons or diffuse gamma-rays (not

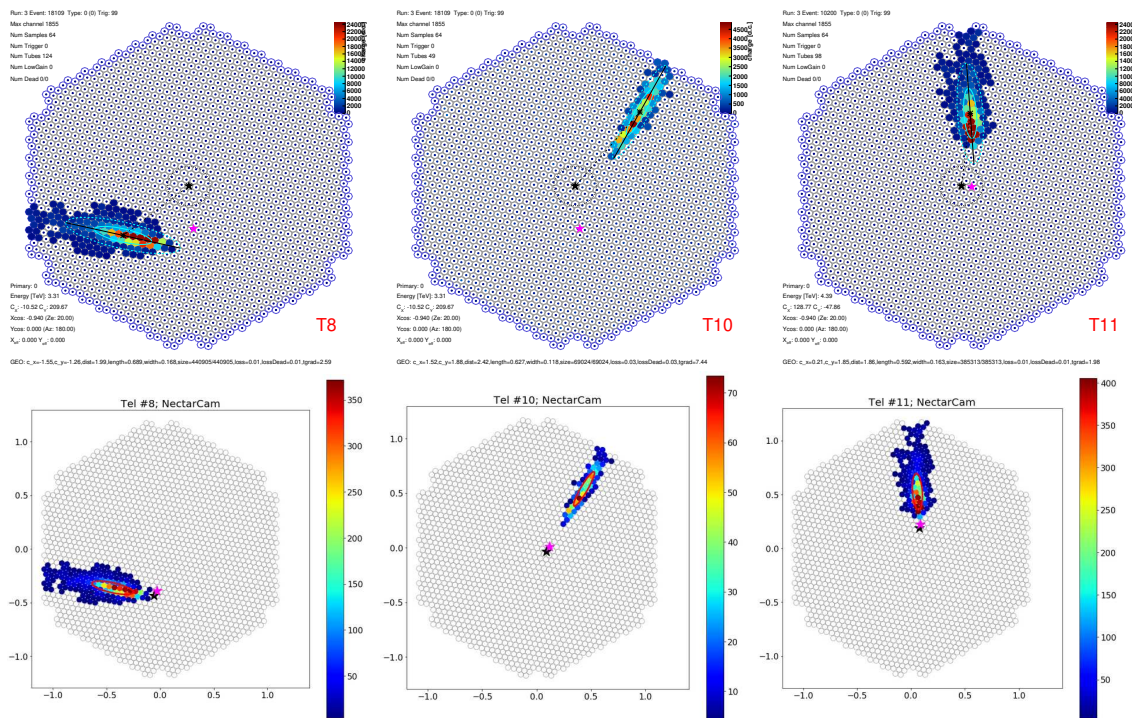


Figure 7.6: Upper panel shows some camera display from EventDisplay, while the lower panel shows the same images, but from ctapipe. The simulation was done using a configuration with a small divergence angle, like the one labeled “0.5”. The black star is the projection of the **simulated** gamma-ray in the camera, while the pink star is the projection of the **reconstructed** sky position in the camera. While EventDisplay does not do the reconstruction properly (the black star is always at the center of the cameras and the reconstruction is wrong), ctapipe does the reconstruction in the right way and the simulated and reconstructed positions are correctly shown in the cameras.

coming from a point source but from an extended region of the sky) that are needed for the final creation of the IRFs.

After some in-house tests, the GRID was used to produce bigger Monte Carlo simulations for the chosen configurations. The choice was motivated by the large disk space and computing power available on GRID and the fact that it was easy to retrieve the data once they have been produced in GRID, after providing the necessary `sim_telarray` configuration files with the modified telescope pointings.

For each of the four different pointing configurations (pointing patterns in figure 7.5) we simulated 1000 files, each with 10^4 showers, each of them re-used 10 times², so that for every configuration we simulated 10^8 gamma-rays in the energy range between 3.0 GeV and 330 TeV coming from a point-like source with a power-law spectrum with an index of -2.0.

Since we managed to obtain the code used for the work done by Lucie Gérard in [103], which is some custom code built around the EventDisplay software package, the first part of the work was to make this custom code work with the new release of EventDisplay and to check the results. At first we tried to analyse some data also with ctapipe and

²In order to save computing time in CORSIKA, each simulated shower can be reused several times, moving the impact point randomly in a box surrounding the array. This is a standard parameter used also in `prod3` with the same value.

EventDisplay in order to have some cross-checks, but due to some difficulties encountered and some missing features in EventDisplay (see figure 7.6), we decided not to spend time in adapting the code in EventDisplay for the divergent mode but to use ctapipe and to adapt it for our analyses using as final cross-checks all the performances plots obtained for the standard EventDisplay analysis with the `prod3` data.

7.1.4 Divergent pointing in ctapipe

The only difference in the analysis of the MC simulations created for the divergent pointing with respect to the standard ones in parallel pointing is at reconstruction level. Since in parallel mode the pointing position is the same for all telescopes, many simplifications can be done when performing the reconstruction, that might not be valid any more in case of divergent pointing.

As already presented in 5.2.1 there are two reconstruction methods available in ctapipe, one fully working in 2 dimensions via a line-intersection method in the common *Nominal frame* and the other one using a 3D intersection method to find the reconstructed position in the sky and the height of the maximum emission of Cherenkov light, plus a 2D method for the reconstruction of the impact point on the ground.

I tested both methods and decided to fully develop the divergent pointing just for the 3D method, since when I started working on this it was easier to use, the code was much easier to read and understand and the performances were expected to be better with respect to the 2D one.

Concerning the 3D reconstruction method, as already said in section 5.2.1, it uses two points in the camera and transforms them to two points in the sky using the transformations between the frame of the camera, the telescope frame and the sky frame.

The two points in the sky, together with the origin, are used to build a plane in a 3D space and since the creation of this plane is done for each telescope separately using its pointing direction, the reconstruction performed using those planes works both in parallel pointing and in the case of divergent pointing (see figure 7.7).

Those planes are then intersected pair-wise, and the angle between them is used as a weight for the computation of the final reconstructed direction, which is a weighed average between all pair-wise directions. Since both the reconstruction of the position in the sky and the height of the maximum of the shower are performed with this 3D method, they both worked for the divergent pointing without any major correction.

Impact point reconstruction

For what concerns the reconstruction of the impact point on the ground, this had to be corrected for the divergent pointing since it was implemented using some approximations that are valid only in standard pointing, but that are no more valid in the divergent pointing mode.

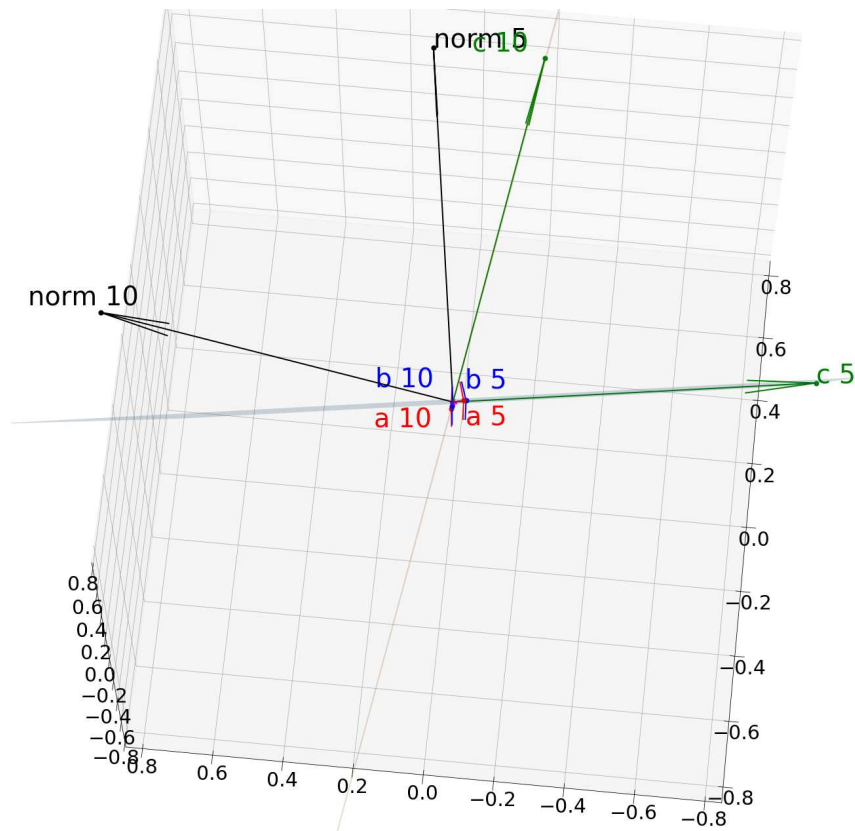


Figure 7.7: display of the 3D reconstruction method for 2 telescopes with IDs 5 and 10. The planes are passing through **a**, **b** and **c**, whereas **norm** is perpendicular to the plane, also drawn here. The planes are viewed from the line resulting from the intersection of these planes, which gives the incoming direction.

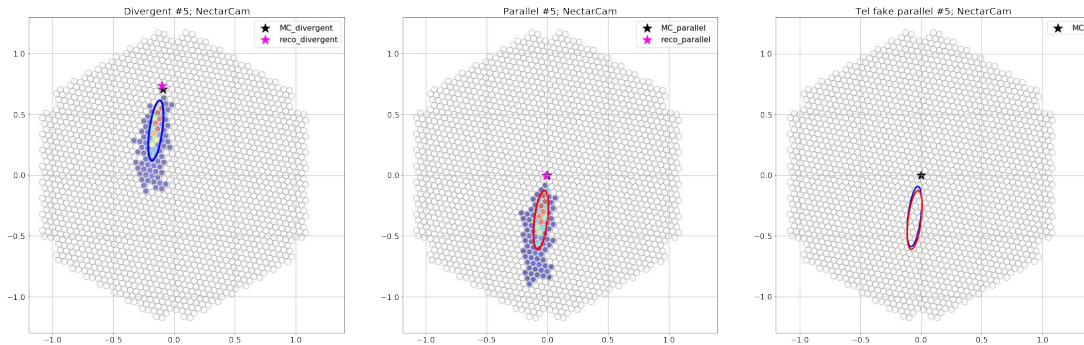


Figure 7.8: the 3 plots shows the camera image (and its ellipse) as observed in divergent pointing mode (left), the same event observed by the same telescope from the simulation in a standard pointing mode (and its ellipse) (centre) and the ellipse resulting after the correction of the reconstruction of the impact point together with the ellipse from the standard pointing (right).

The reconstruction of the impact point on the ground is not done with a 3D method but it's a 2D method which uses a common plane, called "TiltedGroundFrame", perpendicular to the array pointing direction. In the standard mode, the pointing direction of the array coincides with the pointing direction of each telescope, and the camera planes of each telescope are parallel to this common plane. The rotation of the ellipse with respect to the x-axis of the camera, which is one of the Hillas parameters calculated after the cleaning procedure, is used to draw a line which passes through the telescope position and belongs to this reconstruction plane: the intersection of the lines coming from each telescope gives the impact point in the TiltedGroundFrame which is projected into the GroundFrame and can be compared with the impact point from the Monte Carlo simulation which is stored in each event.

Since in the divergent pointing each telescope have a different pointing direction, the assumption of using the rotation angle computed in each camera frame for the reconstruction is not valid any more. This means that the angles that are measured in the cameras in divergent mode cannot be reported on the common plane and used for the impact point reconstruction, as it is done in the parallel pointing case, since angles are not preserved between non-parallel planes and they have to be projected in some way.

In order to properly correct the angle of the ellipse measured in the camera using ctape functions, the two camera points that are projected as points in the sky and used to build the 3D plane for each telescope, are re-projected from the sky to a "fake parallel-pointing" telescope. This is a telescope whose pointing direction coincides with the array pointing direction (it's "fake" because there's no real telescope in the event with such pointing direction). This telescope represents how the same event would have been observed in the camera if the pointing direction of the telescope was the array pointing direction. The goodness of using this reconstruction has been tested using the small Monte Carlo production that we did re-using the CORSIKA simulations with different divergent configurations. I did a comparison event-by-event to check that the parametrization of the ellipse in standard mode and after the correction to the divergent mode were giving

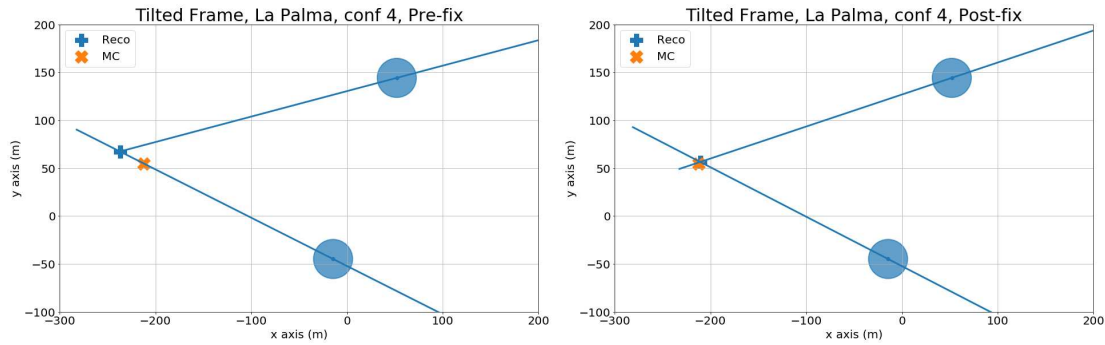


Figure 7.9: Difference between the reconstruction of the impact point before and after the fix, plotted in the tilted ground frame. The event is a 0.96 TeV event which has triggered two MSTs in the divergent configuration 4. The orange cross is the impact point from the **Monte Carlo** whereas the blue cross is the **reconstructed** impact point.

compatible results (see figure 7.8).

After this re-projection the tilt angle of the ellipse has been re-calculated in the camera of this fake telescope and all the corrected tilt angles in the cameras have been used for the impact point reconstruction, showing an improvement in its reconstruction (see figure 7.9).

7.1.5 Results

On-axis The plots³ on the left in figure 7.10 represent the angular resolution that we have obtained for the four configurations that we have been investigating (pointings are in figure 7.5). The plots represent the angular separation between the reconstructed position in the sky and the true position of the source, with the error bar representing 68% of the events in that energy bin. These plots have been published in [104].

The angular resolution plots usually represent the angle within which 68% of reconstructed gamma rays fall, relative to their true direction and are plotted against the reconstructed energy: gamma-hadron separation cuts are usually applied to the MC that is being used to determine the angular resolution. In our analyses, we have not taken into account the protons but only point gamma-rays, so no gamma-hadron separation have been carried out and the angular resolution is plotted against the true energy from the simulation.

The plots for the on-axis configurations in figure 7.10 were obtained applying some selection cuts: the resulting image after the cleaning must have at least 6 pixels left and the leakage, which is the percentage of signal deposited in the border of the camera, had to be lower than 10%. Moreover, the picture threshold, the boundary and the minimum number of neighbours in the tailcut cleaning were 5, 10 and 2 respectively (see section 5.1.2 on cleaning).

The results that we have obtained show that a more divergent configuration gives a worse angular resolution with respect to a less divergent one, especially at high energies and with low multiplicity. A higher multiplicity gives a better angular resolution over the

³All the angular resolution plots have been created with ctaplot (<https://github.com/vuillaud/ctaplot>).

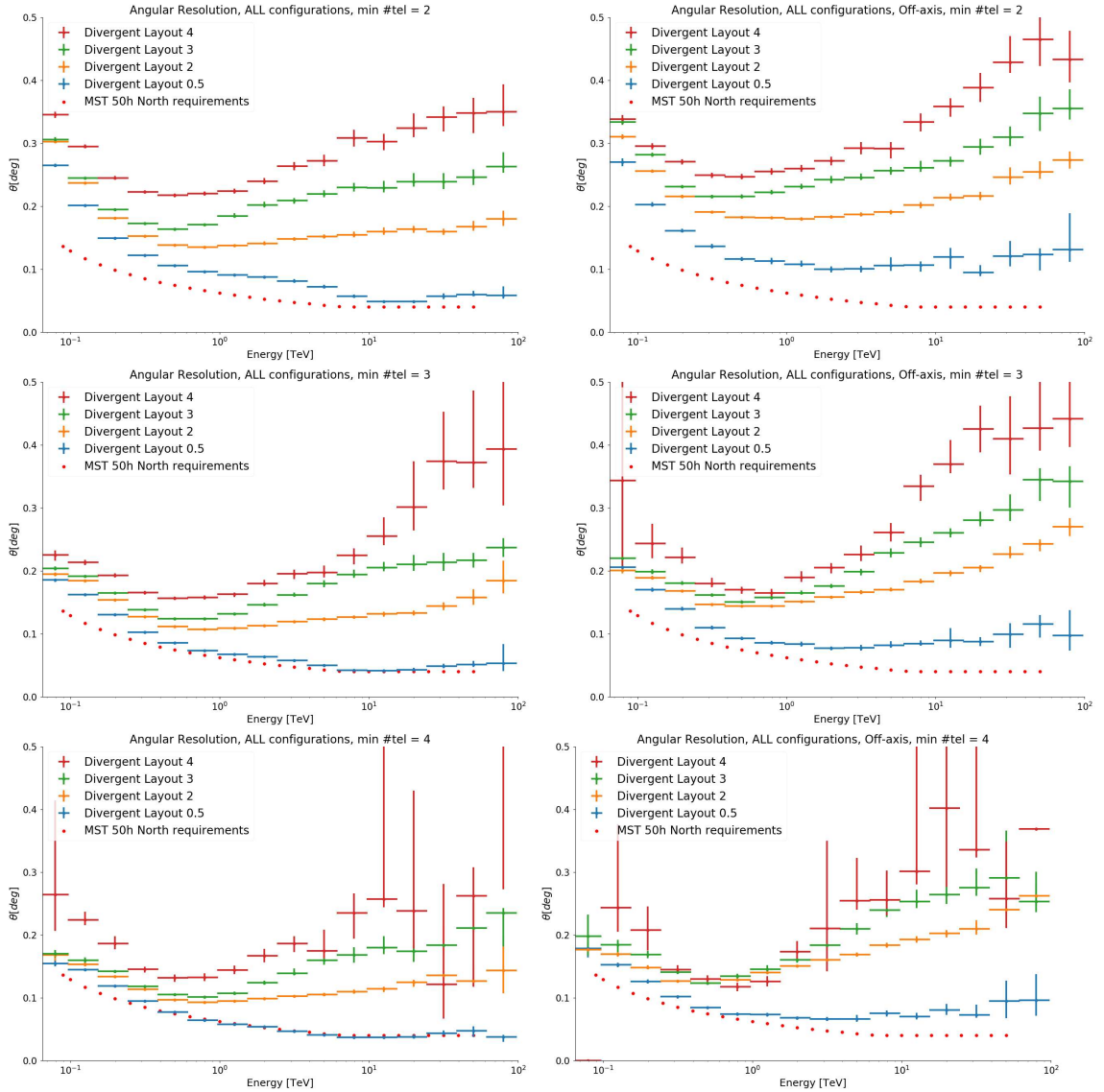


Figure 7.10: Angular resolution plots for the four divergent pointing configurations tested, plotted against the requirements for the MST on the North. Left is the on-axis configuration, while right is the off-axis configuration (see the off-axis result section for the details). The different rows have a different minimum number of triggered telescopes, from 2 for the first row to 4 for the last row. The on-axis plots have been presented at the 2019 ICRC Conference and published as a proceeding [104].

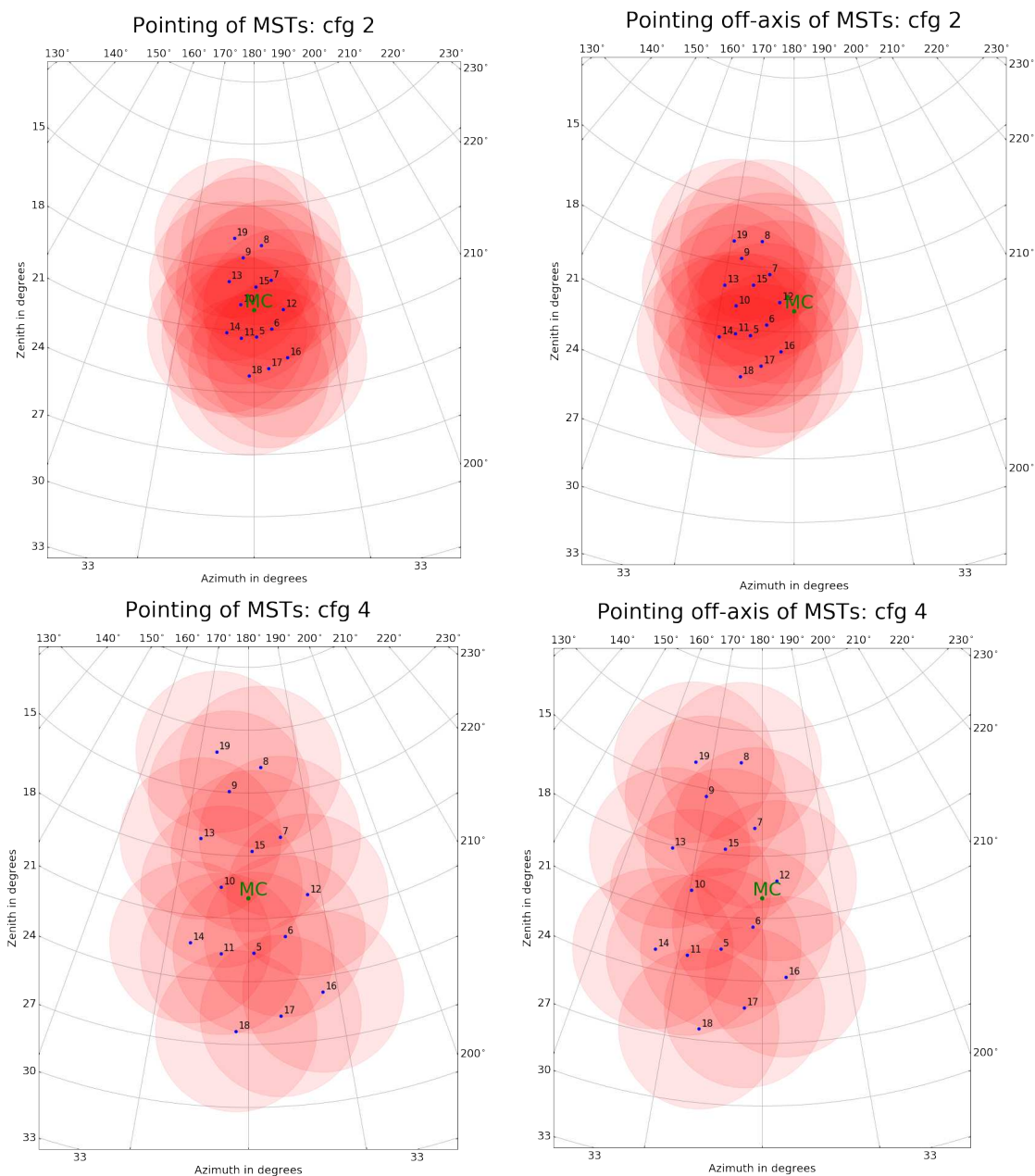


Figure 7.11: the plots show the difference in the pointing directions of the MSTs between the on-axis simulations (right) and the off-axis simulation (left). While the direction of the gamma-rays is the same for both simulations, the average pointing direction of the telescopes are different. *Top*: plots are for the layout “2”. *Bottom*: plots are for the layout “4”.

whole energy range, at the expense of a lower number of events; the configuration labelled “0.5”, very similar to the parallel pointing configuration, has an angular resolution close to the requirements, as expected, while there is a small difference between the layouts “2” and “3” below 1 TeV.

Off-axis Together with the configurations shown in figure 7.5 in which the average pointing of the telescopes is the position from which the gamma-rays are coming, we have also tested a similar set of configurations, called “off-axis”. In this second set of configurations,

the telescopes' pointings are spread around an azimuth of 174 deg and an altitude of 70 deg (see on-axis vs off-axis difference in figure 7.11), with the gamma-rays coming from 180 deg of azimuth and 70 of altitude (results on the right side in figure 7.10). The results for those "off-axis" configurations do not differ that much from those presented for the "on-axis" case, since the difference in azimuth with respect to the "on-axis" configurations is 6° , but the angular separation at 70° of altitude is reduced to $6^\circ \times \cos(70^\circ) = 2.05^\circ$, which is roughly 1/4th of the field of view of the MST.

The difference between "on-axis" and "off-axis" configurations is more pronounced for layout that are characterized by a low divergent angle (such as the "0.5" and the "2") while this difference is smaller for the layout "4" with a large divergence angle.

7.1.6 Conclusions

The preliminary results shown here are a first step in the full divergent pointing analysis, whose final goal is to produce IRFs for a divergent configuration. The IRF production will be done using a pipeline called **protopipe** (<https://github.com/cta-observatory/protopipe>) which has been developed by the group in Saclay and has been released at the beginning of June 2019.

The **protopipe** pipeline is built on top of **ctapipe** and the aim is to use it to produce the IRFs from the future Monte Carlo productions, instead of using **EventDisplay**. The pipeline is not fully implemented yet and the results that are produced are not completely in agreement with those that are obtained from the **EventDisplay** or **MARS** analysis chains: a deep cross-check between the **ctapipe** and the other software packages is being carried on in the Analysis and Simulation Working Group in order to make sure that the results that are produced by **protopipe** are right and the old software packages can be abandoned in favour of this new pipeline.

We are working together with the group developing **protopipe** to help in the debugging of the analysis and to make sure that it can be used in the near future to produce IRFs for a divergent pointing configuration. Both the standard analysis and our divergent pointing analysis will have micro-optimizations to do in order to have a much fair comparison with the results that have been produced so far with the other software packages.

The results presented in this work have shown only the angular resolution plots created from a simulation of pure gamma-rays. No energy resolution plot or effective area has been produced for this first publication since the focus was mainly on the direction reconstruction and the angular resolution. An update of this work will include also the analysis of the proton simulations with energy resolution and effective areas, all computer using the **protopipe** pipeline.

Ctapipe is also not complete and it is in a beta version, with big modifications to the code still happening but less modifications affecting the usage of **ctapipe** classes and methods in an analysis pipeline.

Chapter 8

Conclusions

The thesis has been developed in the context of the Cherenkov Telescope Array, the major next-generation observatory for ground-based very-high-energy gamma-ray astronomy. The work has covered a broad range of topics, from the low-level analysis of the raw data, to the high-level simulations using the Instrument Response Functions to determine the capabilities of CTA towards the observations of several astrophysical sources.

The low-level analysis work has been carried around the software library **ctapipe**, with a double goal. The first goal was to contribute to the development of the framework, debugging the code and writing the unit tests, participate to the discussion for what concerns the implementation of new features and solve the issues presented by the users while helping them in using the framework. The second goal was to test the framework on some dedicated analysis, such as the “divergent pointing” mode and the ImPACT analysis, in order to implement some new features into **ctapipe**. The preliminary work done for the divergent pointing mode, which has brought some major changes into the framework, is the most up-to-date work concerning this new type of pointing and it is considered a very important item both in the Analysis and Simulation Working Group of the CTA Consortium and at the level of the CTA Observatory. It will be important to know the performances of the array with the divergent mode in order to make it a feasible observing mode for future external scientific proposals being carried on by the Observatory. The ImPACT analysis has been used by few people inside the collaboration so far due to a missing public dataset for the templates to use in the fitting procedure, but due to the interest in porting this analysis of GPU with PyTorch, a work which needs further attention and more time, I can easily understand the implementation, run the analysis and use it to get some meaningful results. ImPACT, due to the performances shown in the analysis of the H.E.S.S. data, is an advanced analysis and it will be important to continue to implement it and test it inside **ctapipe** in order to improve the performances of CTA: a first important test will be related to the usage of such method in the analysis of the data collected by the LST-1.

The high-level analysis pipelines have been used for many different topics, from the First CTA Data Challenge (DC-1) to the Consortium papers on the detection of Gamma-

Ray Bursts with CTA and on the follow-up of Gravitational Waves events. The contribution to the DC-1 for what concerns the creation of the population of AGNs, was an important task, requested by the Extragalactic Working Group of the Consortium, that I took care of, learning where to look for data and how to handle data coming from different catalogues. In order to complete this work, I had to write several script to automatize the search for sources in the catalogues and the actual creation of the XML files to be used by the group responsible for the creation of the simulations using those input models. This work, and the DC-1 in general, has been really important inside the Consortium to involve more people in the analysis of the CTA data with high-level tools and to test them. The results presented at the end of the DC-1 will be used for the Second CTA Data Challenge, in which more advanced models for the sources will be included and which will be developed according to the needs of the CTA Observatory.

The work on the study of the performances of CTA towards the observation of GRBs went through different steps during the thesis, from a more technical initial work using simple physical models to test the IRFs and all the various technical details, to the recent work done for the Consortium paper. The expertise accumulated over these years regarding the development of computing pipelines in Python and the knowledge of the high-level analysis tools, has driven the development of the pipeline to be used for the Consortium paper on GRBs and the one on the follow-up of Gravitational Wave events.

Appendix A

The Extragalactic Background Light

The integrated emission from stars and galaxies of all type through the evolution of the universe, called Extragalactic Background Light (EBL), is the second-most-intense form of cosmic background light the first being the cosmic microwave background (see figure A.1).

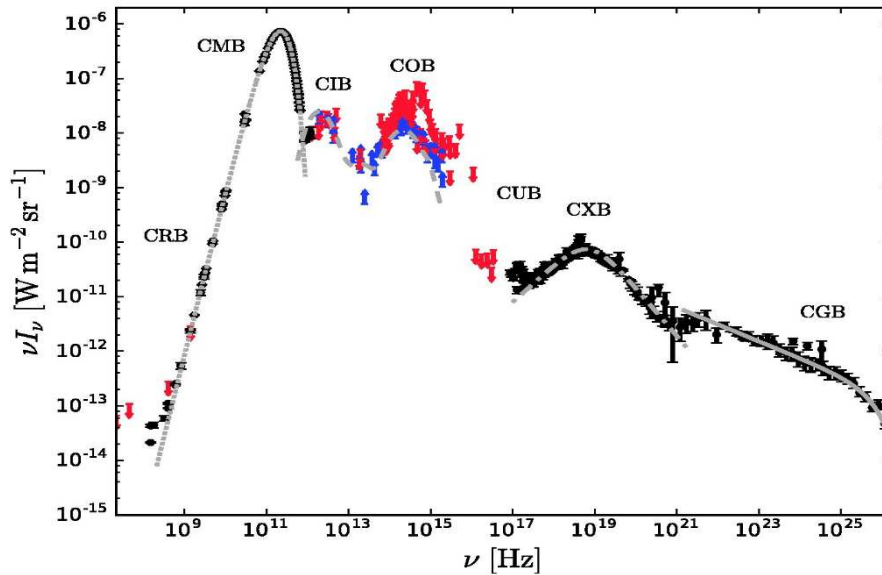


Figure A.1: The figure (from [105]) shows the backgrounds at different wavelengths. The Cosmic Microwave Background is the most intense, while the optical (COB) and infrared (CIB) are the second-most-important.

This light, in the optical, ultraviolet and near-infrared wavelengths, consists of the integrated light, averaged over a 4π solid angle, of all galaxies in the universe plus any contribution by individual stars or emission by gas outside the galaxies. Furthermore, there might be a significant number of galaxies with a low surface brightness that cannot be observed as discrete objects but that are contributing to the cumulative EBL.

A fraction of these photons is absorbed by interstellar dust and re-emitted at longer

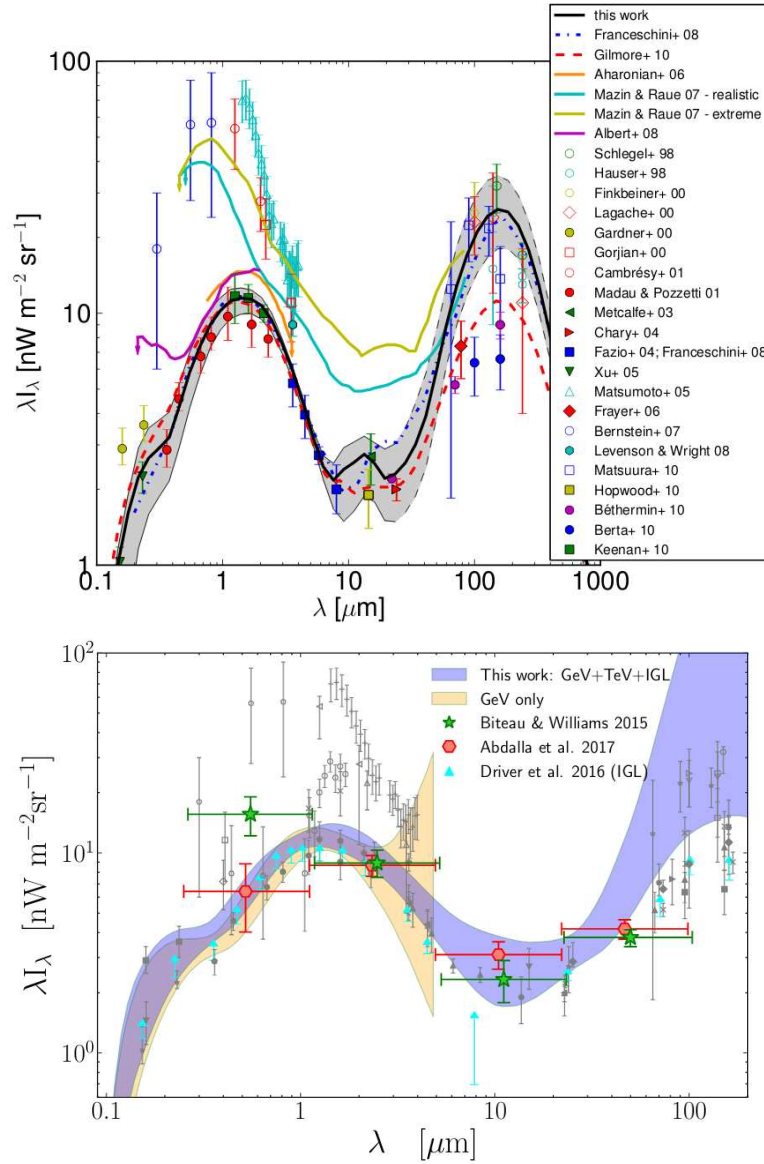


Figure A.2: *Upper panel:* the figure (from [82]) shows the EBL model obtained by Dominguez in 2011, compared with other previous models. The characteristics two bumps can be seen at $\lambda \approx 1 \mu\text{m}$ and $\lambda \approx 100 \mu\text{m}$. *Lower panel:* the figure (from [106]) shows the result on the combined analysis done with data from Fermi and MAGIC for a sample of 12 blazars with redshift up to 1.

wavelengths, producing the characteristic double peak spectral energy distribution of the EBL (see figure A.2).

While it is very difficult to measure directly due to the strong foreground contamination from the Solar System and the Milky Way (see for example [105] and [106]), the signature left on the flux of sources at large redshift, due to interaction of this IR–UV light with the VHE photons, can be used to model the EBL.

A precise modelling of the EBL from the observation of source at energies above 10 GeV from space or with IACTs [106], is used to understand the star formation history of the universe or to probe the very weak magnetic fields in cosmic voids down to values

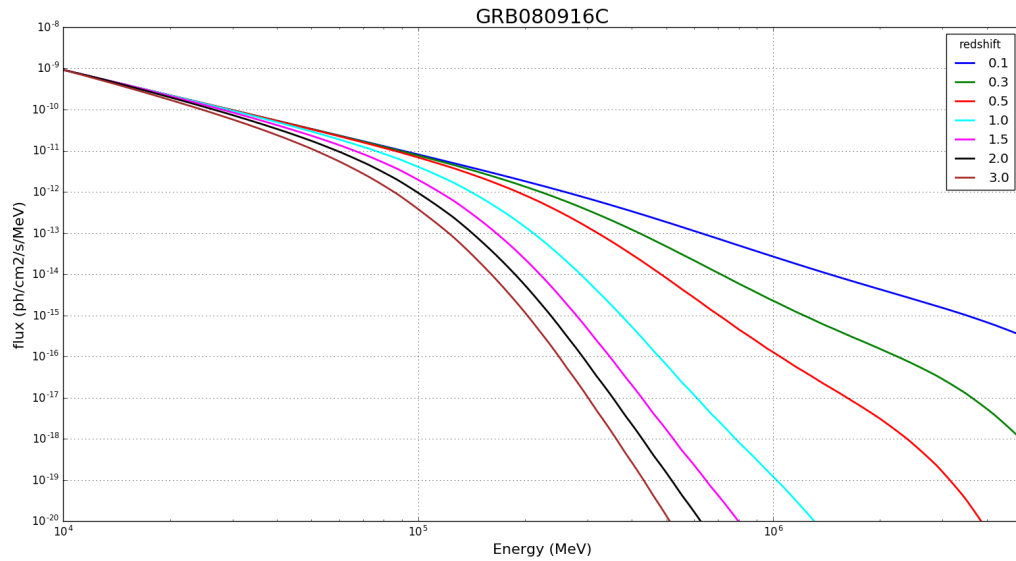


Figure A.3: Effect of the EBL on the model of GRB080916C when this source is placed at different redshifts. The model used here is the one from Franceschini [76].

many orders of magnitude below any other technique can reach.

Even though the effect of the EBL on the model of this source is a decrease of the intrinsic flux at the highest energies, the effect being higher with increasing energy and increasing redshift (see figure A.3), it's important to have a very precise knowledge of the EBL model. While observing a source at the highest energies, a deviation from the expected spectrum of the source after the EBL has been considered, can be a hint of new physics (see chapter 2 and [107]).

Appendix B

CTAO hierarchical data model

The different data levels in data processing range from the raw level data to the high-level data. I'll list here the data levels defined by CTAO¹ leaving specific details and examples to the following sections:

R0 (raw low-level) this data level is the lowest level and it's content and format is internal to each device/system. This level represents the on-line streamed raw data, not normally preserved, and are directly streamed from the telescopes to the on-site Data Centre.

R1 (raw common) this data level meets some standards and it's meant to be streamed to other systems belonging to the ACADA (Array Control and Data Acquisition system of CTA) and it's the first data level seen by this system: the creation of this data level will need some pre-processing from the R0 data level.

DL0 (raw archived) all archival data from data acquisition hardware/software which represents the rawest data level intended for long-term storage. This data level doesn't represent only some calibrated data from the cameras but also measurements from other systems and softwares (i.e. instantaneous measurements from drive system).

DL1 (processed) per-telescope data either connected to measurements relative to telescopes subsystems (i.e. average measurements from engines or positions) or to the camera. The calibrated image charge (.IMAGE data sub-type) and the Hillas parametrization (.PARAM data sub-type) fall in this data level.

DL2 (reconstructed) until now all the informations were basically telescope-wise or site-wise, whereas the DL2 is the data level specific for reconstructed parameters from the observation of an event done with the telescopes belonging to a subarray (that have passed previous cleanings and selections cuts).

¹From the "CTA Data Model Identification and Core Metadata Schema" document. This is an internal document available only to CTA members.

DL3 (science) the DL3 is a list a Air-Shower Events with a single final set of reconstruction and discrimination parameters (stored as a simple table, with one event in each row) which comes together with all the necessary informations for an user to perform the scientific analysis, such as the IRF for a specific time bin or any other technical data. There might be some quality cuts applied to this level according to the science goal which is being investigated.

References

- [1] S. Mollerach and E. Roulet. “Progress in high-energy cosmic ray physics”. In: *Prog. Part. Nucl. Phys.* 98 (2018), pp. 85–118. DOI: [10.1016/j.ppnp.2017.10.002](https://doi.org/10.1016/j.ppnp.2017.10.002). arXiv: 1710.11155 [astro-ph.HE].
- [2] A. M. Hillas. “Cherenkov light images of EAS produced by primary gamma”. In: *International Cosmic Ray Conference 3* (Aug. 1985).
- [3] E. Aliu et al. “Improving the performance of the single-dish Cherenkov telescope MAGIC through the use of signal timing”. In: *Astropart. Phys.* 30 (2009), pp. 293–305. DOI: [10.1016/j.astropartphys.2008.10.003](https://doi.org/10.1016/j.astropartphys.2008.10.003). arXiv: 0810.3568 [astro-ph].
- [4] Werner Hofmann. “Perspectives from CTA in relativistic astrophysics”. In: *Int. J. Mod. Phys. D* 26.03 (2016), p. 1730005. DOI: [10.1142/S0218271817300051](https://doi.org/10.1142/S0218271817300051).
- [5] A. Acharyya et al. “Monte Carlo studies for the optimisation of the Cherenkov Telescope Array layout”. In: *Astroparticle Physics* 111 (2019), pp. 35–53. ISSN: 0927-6505. DOI: <https://doi.org/10.1016/j.astropartphys.2019.04.001>. URL: <http://www.sciencedirect.com/science/article/pii/S0927650519300234>.
- [6] L Arrabito et al. “The Cherenkov Telescope Array production system for Monte Carlo simulations and analysis”. In: *Journal of Physics: Conference Series* 898 (2017), p. 052013. DOI: [10.1088/1742-6596/898/5/052013](https://doi.org/10.1088/1742-6596/898/5/052013). URL: <https://doi.org/10.1088/1742-6596/898/5/052013>.
- [7] URL: <https://www.cta-observatory.org/science/cta-performance/>.
- [8] Konrad Bernlöhr. “Simulation of imaging atmospheric Cherenkov telescopes with CORSIKA and sim_telarray”. In: *Astroparticle Physics* 30.3 (2008), pp. 149–158. ISSN: 0927-6505. DOI: <https://doi.org/10.1016/j.astropartphys.2008.07.009>. URL: <http://www.sciencedirect.com/science/article/pii/S0927650508000972>.
- [9] Roberta Zanin. “MARS, the MAGIC analysis and reconstruction software”. In: *Proceedings, 33rd International Cosmic Ray Conference (ICRC2013): Rio de Janeiro, Brazil, July 2-9, 2013*, p. 0773. URL: <http://www.cbpf.br/~7Eicrc2013/papers/icrc2013-0773.pdf>.

- [10] Gernot Maier and Jamie Holder. “Eventdisplay: An Analysis and Reconstruction Package for Ground-based Gamma-ray Astronomy”. In: *PoS ICRC2017* (2018). [35,747(2017)], p. 747. DOI: [10.22323/1.301.0747](https://doi.org/10.22323/1.301.0747). arXiv: [1708.04048](https://arxiv.org/abs/1708.04048) [astro-ph.IM].
- [11] Maxim Shayduk for the CTA Consortium. “Optimized next-neighbor image cleaning method for Cherenkov Telescopes”. In: *arXiv e-prints*, arXiv:1307.4939 (2013), arXiv:1307.4939. arXiv: [1307.4939](https://arxiv.org/abs/1307.4939) [astro-ph.IM].
- [12] C. W. Akerlof et al. “Locating Very High Energy Gamma-Ray Sources with Arcminute Accuracy”. In: *Astrophys. J.* 377 (1991), p. L97. DOI: [10.1086/186125](https://doi.org/10.1086/186125).
- [13] E. Domingo-Santamaria et al. “The DISP analysis method for point-like or extended gamma source searches/studies with the MAGIC Telescope”. In: *29th International Cosmic Ray Conference (ICRC29), Volume 5*. Vol. 5. International Cosmic Ray Conference. 2005, p. 363.
- [14] C. C. Lu. *Improving the H.E.S.S. angular resolution using the Disp method*. 2013. arXiv: [1310.1200](https://arxiv.org/abs/1310.1200) [astro-ph.IM].
- [15] Gunes Demet Senturk. “The Disp Method for Analysing Large Zenith Angle Gamma-Ray Data”. In: *International Cosmic Ray Conference*. Vol. 9. International Cosmic Ray Conference. 2011, p. 127. DOI: [10.7529/ICRC2011/V09/0925](https://doi.org/10.7529/ICRC2011/V09/0925). arXiv: [1109.6044](https://arxiv.org/abs/1109.6044) [astro-ph.IM].
- [16] J. Albert et al. “Implementation of the Random Forest method for the Imaging Atmospheric Cherenkov Telescope MAGIC”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 588 (2008), pp. 424–432. DOI: [10.1016/j.nima.2007.11.068](https://doi.org/10.1016/j.nima.2007.11.068). URL: <http://dx.doi.org/10.1016/j.nima.2007.11.068>.
- [17] H. Krawczynski et al. “Gamma-hadron separation methods for the VERITAS array of four imaging atmospheric Cherenkov telescopes”. In: *Astroparticle Physics* 25.6 (2006), pp. 380–390. ISSN: 0927-6505. DOI: <https://doi.org/10.1016/j.astropartphys.2006.03.011>. URL: <http://www.sciencedirect.com/science/article/pii/S0927650506000466>.
- [18] J. Albert et al. “Implementation of the Random Forest method for the Imaging Atmospheric Cherenkov Telescope MAGIC”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 588.3 (2008), pp. 424–432. ISSN: 0168-9002. DOI: <https://doi.org/10.1016/j.nima.2007.11.068>. URL: <http://www.sciencedirect.com/science/article/pii/S0168900207024059>.
- [19] H.S. Ahn et al. “The energy spectra of protons and helium measured with the ATIC experiment”. In: *Advances in Space Research* 37.10 (2006). Astrophysics, pp. 1950–1954. ISSN: 0273-1177. DOI: <https://doi.org/10.1016/j.asr.2005.09.031>. URL: <http://www.sciencedirect.com/science/article/pii/S0273117705011233>.

- [20] F. Aharonian et al. “The Crab Nebula and Pulsar between 500 GeV and 80 TeV: Observations with the HEGRA Stereoscopic Air Cerenkov Telescopes”. In: *Astrophys. J.* 614.2 (2004), pp. 897–913. DOI: [10.1086/423931](https://doi.org/10.1086/423931). arXiv: [astro-ph/0407118](https://arxiv.org/abs/astro-ph/0407118) [astro-ph].
- [21] Maria Krause, Elisa Pueschel, and Gernot Maier. “Improved γ /hadron separation for the detection of faint γ -ray sources using boosted decision trees”. In: *Astroparticle Physics* 89 (2017), pp. 1–9. ISSN: 0927-6505. DOI: <https://doi.org/10.1016/j.astropartphys.2017.01.004>. URL: <http://www.sciencedirect.com/science/article/pii/S0927650517300166>.
- [22] T.-P. Li and Y.-Q. Ma. “Analysis methods for results in gamma-ray astronomy”. In: *Astrophys. J.* 272 (Sept. 1983), pp. 317–324. DOI: [10.1086/161295](https://doi.org/10.1086/161295).
- [23] Daniel Mazin. “The Cherenkov Telescope Array”. In: *36th International Cosmic Ray Conference (ICRC 2019) Madison, Wisconsin, USA, July 24-August 1, 2019*. 2019. arXiv: [1907.08530](https://arxiv.org/abs/1907.08530) [astro-ph.IM].
- [24] B. S. Acharya et al. *Science with the Cherenkov Telescope Array*. WSP, 2018. ISBN: 9789813270084. DOI: [10.1142/10986](https://doi.org/10.1142/10986). arXiv: [1709.07997](https://arxiv.org/abs/1709.07997) [astro-ph.IM].
- [25] Carmelo Evoli. *The Cosmic-Ray Energy Spectrum*. Oct. 2018. DOI: [10.5281/zenodo.2360277](https://doi.org/10.5281/zenodo.2360277). URL: <https://doi.org/10.5281/zenodo.2360277>.
- [26] Frank G. Schröder. *News from Cosmic Ray Air Showers (ICRC 2019 – Cosmic Ray Indirect Rapport)*. 2019. arXiv: [1910.03721](https://arxiv.org/abs/1910.03721) [astro-ph.HE].
- [27] M. et al. Ackermann. “Detection of the Characteristic Pion-Decay Signature in Supernova Remnants”. In: *Science* 339.6121 (2013), pp. 807–811. ISSN: 0036-8075. DOI: [10.1126/science.1231160](https://doi.org/10.1126/science.1231160). URL: <https://science.sciencemag.org/content/339/6121/807>.
- [28] F. Acero et al. “Prospects for Cherenkov Telescope Array Observations of the Young Supernova Remnant RX J1713.7-3946”. In: *The Astrophysical Journal* 840.2 (2017), p. 74. ISSN: 1538-4357. DOI: [10.3847/1538-4357/aa6d67](https://doi.org/10.3847/1538-4357/aa6d67). URL: <http://dx.doi.org/10.3847/1538-4357/aa6d67>.
- [29] C. Pfrommer et al. “Cosmic ray feedback in galaxies and active galactic nuclei”. In: *6th International Symposium on High Energy Gamma-Ray Astronomy*. Vol. 1792. American Institute of Physics Conference Series. 2017, p. 030003. DOI: [10.1063/1.4968904](https://doi.org/10.1063/1.4968904).
- [30] H. Sol et al. “Active Galactic Nuclei under the scrutiny of CTA”. In: *Astroparticle Physics* 43 (2013). Seeing the High-Energy Universe with the Cherenkov Telescope Array - The Science Explored with the CTA, pp. 215–240. ISSN: 0927-6505. DOI: <https://doi.org/10.1016/j.astropartphys.2012.12.005>. URL: <http://www.sciencedirect.com/science/article/pii/S0927650512002228>.

- [31] Robert Parsons et al. “Sgr A* Observations with H.E.S.S. II”. In: *PoS ICRC2015* (2016). [34,830(2015)], p. 830. DOI: 10.22323/1.236.0830. arXiv: 1509.03425 [astro-ph.HE].
- [32] P. A. R. Ade et al. “Planck 2015 results. XIII. Cosmological parameters”. In: *Astron. Astrophys.* 594 (2016), A13. DOI: 10.1051/0004-6361/201525830. arXiv: 1502.01589 [astro-ph.CO].
- [33] John Ellis et al. “Robust Constraint on Lorentz Violation Using Fermi-LAT Gamma-Ray Burst Data”. In: *Phys. Rev. D* 99.8 (2019), p. 083009. DOI: 10.1103/PhysRevD.99.083009. arXiv: 1807.00189 [astro-ph.HE].
- [34] Asaf Pe’er. “Physics of Gamma-Ray Bursts Prompt Emission”. In: *Adv. Astron. 2015* (2015), p. 907321. DOI: 10.1155/2015/907321. arXiv: 1504.02626 [astro-ph.HE].
- [35] C. Kouveliotou et al. “Identification of two classes of gamma-ray bursts”. In: *Astrophys. J. Lett.* 413 (Aug. 1993), pp. L101–L104. DOI: 10.1086/186969.
- [36] William S. Paciesas et al. “The Fourth BATSE Gamma-Ray Burst Catalog (Revised)”. In: *The Astrophysical Journal Supplement Series* 122.2 (1999), p. 465. URL: <http://stacks.iop.org/0067-0049/122/i=2/a=465>.
- [37] C. A. Meegan et al. “Spatial distribution of gamma-ray bursts observed by BATSE”. In: *Nature* 355 (Jan. 1992), pp. 143–145. DOI: 10.1038/355143a0.
- [38] Pawan Kumar and Bing Zhang. “The physics of gamma-ray bursts & relativistic jets”. In: *Phys. Rept.* 561 (2014), pp. 1–109. DOI: 10.1016/j.physrep.2014.09.008. arXiv: 1410.0679 [astro-ph.HE].
- [39] E. Costa et al. “Discovery of an X-ray afterglow associated with the gamma-ray burst of 28 February 1997”. In: *Nature* 387 (1997), pp. 783–785. DOI: 10.1038/42885. arXiv: astro-ph/9706065 [astro-ph].
- [40] M. Galli et al. “AGILE mini-calorimeter gamma-ray burst catalog”. In: *Astronomy & Astrophysics* 553 (2013), A33. ISSN: 1432-0746. DOI: 10.1051/0004-6361/201220833. URL: <http://dx.doi.org/10.1051/0004-6361/201220833>.
- [41] P. Narayana Bhat et al. “The Third Fermi gbm Gamma-ray Burst Catalog: the First six Years”. In: *Astrophys. J. Suppl.* 223.2 (2016), p. 28. DOI: 10.3847/0067-0049/223/2/28. arXiv: 1603.07612 [astro-ph.HE].
- [42] M. Ajello et al. “A Decade of Gamma-Ray Bursts Observed by Fermi-LAT: The Second GRB Catalog”. In: *The Astrophysical Journal* 878.1 (2019), p. 52. DOI: 10.3847/1538-4357/ab1d4e. URL: <https://doi.org/10.3847%2F1538-4357%2Fab1d4e>.
- [43] D. Band et al. “BATSE observations of gamma-ray burst spectra. I - Spectral diversity”. In: *Astroph. J.* 413 (Aug. 1993), pp. 281–292. DOI: 10.1086/172995.

- [44] “Fermi Observations of High-Energy Gamma-Ray Emission from GRB 080916C”. In: *Science* 323.5922 (2009). Ed. by A. A. et al. Abdo, pp. 1688–1693. ISSN: 0036-8075. DOI: [10.1126/science.1169101](https://doi.org/10.1126/science.1169101). URL: <http://science.sciencemag.org/content/323/5922/1688>.
- [45] P. Mészáros and M. J. Rees. “Optical and Long-Wavelength Afterglow from Gamma-Ray Bursts”. In: *The Astrophysical Journal* 476.1 (1997), p. 232. URL: <http://stacks.iop.org/0004-637X/476/i=1/a=232>.
- [46] E. Waxman, S. R. Kulkarni, and D. A. Frail. “Implications of the Radio Afterglow from the Gamma-Ray Burst of 1997 May 8”. In: *The Astrophysical Journal* 497.1 (1998), 288–293. ISSN: 1538-4357. DOI: [10.1086/305467](https://doi.org/10.1086/305467). URL: <http://dx.doi.org/10.1086/305467>.
- [47] G. Cavallo and M. J. Rees. “A qualitative study of cosmic fireballs and γ -ray bursts”. In: *Monthly Notices of the Royal Astronomical Society* 183.3 (1978), pp. 359–365. DOI: [10.1093/mnras/183.3.359](https://doi.org/10.1093/mnras/183.3.359). URL: <http://mnras.oxfordjournals.org/content/183/3/359.abstract>.
- [48] N. Gehrels and P. Mészáros. “Gamma-Ray Bursts”. In: *Science* 337 (Aug. 2012), p. 932. DOI: [10.1126/science.1216793](https://doi.org/10.1126/science.1216793). arXiv: 1208.6522 [astro-ph.HE].
- [49] P. Kumar et al. “Maximum synchrotron frequency for shock-accelerated particles”. In: *Monthly Notices of the Royal Astronomical Society: Letters* 427.1 (2012), pp. L40–L44. DOI: [10.1111/j.1745-3933.2012.01341.x](https://doi.org/10.1111/j.1745-3933.2012.01341.x). URL: <http://mnrasl.oxfordjournals.org/content/427/1/L40.abstract>.
- [50] V. Acciari et al. “Teraelectronvolt emission from the γ -ray burst GRB 190114C”. In: *Nature* 575 (2019), pp. 455–458. DOI: [10.1038/s41586-019-1750-x](https://doi.org/10.1038/s41586-019-1750-x).
- [51] P. Crumley and P. Kumar. “Hadronic models for Large Area Telescope prompt emission observed in Fermi gamma-ray bursts”. In: *Monthly Notices of the Royal Astronomical Society* 429.4 (2013), pp. 3238–3251. ISSN: 0035-8711. DOI: [10.1093/mnras/sts581](https://doi.org/10.1093/mnras/sts581). URL: <https://doi.org/10.1093/mnras/sts581>.
- [52] Lara Nava. “High-energy emission from Gamma-Ray Bursts”. In: *Int. J. Mod. Phys. D* 27.13 (2018), p. 1842003. DOI: [10.1142/S0218271818420038](https://doi.org/10.1142/S0218271818420038). arXiv: 1804.01524 [astro-ph.HE].
- [53] M. Ajello et al. “The First Fermi LAT Gamma-Ray Burst Catalog”. In: *Astrophys. J. Suppl.* 209.1 (2013), p. 11. DOI: [10.1088/0067-0049/209/1/11](https://doi.org/10.1088/0067-0049/209/1/11). arXiv: 1303.2908 [astro-ph.HE].
- [54] Asaf Pe’er. “Physics of Gamma-Ray Bursts Prompt Emission”. In: *Adv. Astron.* 2015 (2015), p. 907321. DOI: [10.1155/2015/907321](https://doi.org/10.1155/2015/907321). arXiv: 1504.02626 [astro-ph.HE].
- [55] T.C. Weekes. *Very high energy gamma-ray astronomy*. 1st ed. Series in astronomy and astrophysics. Institute of Physics Pub, Bristol, UK, 2003. ISBN: 9780750306584.

- [56] R. Mirzoyan. “First time detection of a GRB at sub-TeV energies; MAGIC detects the GRB 190114C”. In: *The Astronomer’s Telegram* 12390 (Jan. 2019).
- [57] H. Abdalla et al. “A very-high-energy component deep in the γ -ray burst afterglow”. In: *Nature* 575.7783 (2019), pp. 464–467. ISSN: 1476-4687. DOI: [10.1038/s41586-019-1743-9](https://doi.org/10.1038/s41586-019-1743-9). URL: <http://dx.doi.org/10.1038/s41586-019-1743-9>.
- [58] M. de Naurois. “GRB190829A: Detection of VHE gamma-ray emission with H.E.S.S.” In: *The Astronomer’s Telegram* 13052 (Aug. 2019).
- [59] B. P. Abbott et al. “Gravitational Waves and Gamma-Rays from a Binary Neutron Star Merger: GW 170817 and GRB 170817A”. In: *The Astrophysical Journal* 848.2 (2017), p. L13. DOI: [10.3847/2041-8213/aa920c](https://doi.org/10.3847/2041-8213/aa920c). URL: <https://doi.org/10.3847/2041-8213/aa920c>.
- [60] Susumu Inoue et al. “Gamma-ray burst science in the era of the Cherenkov Telescope Array”. In: *Astroparticle Physics* 43 (2013), pp. 252–275. ISSN: 0927-6505. DOI: [10.1016/j.astropartphys.2013.01.004](https://doi.org/10.1016/j.astropartphys.2013.01.004). URL: <http://dx.doi.org/10.1016/j.astropartphys.2013.01.004>.
- [61] V. Acciari et al. “Observation of inverse Compton emission from a long γ -ray burst”. In: *Nature* 575 (2019), pp. 459–463. DOI: [10.1038/s41586-019-1754-6](https://doi.org/10.1038/s41586-019-1754-6).
- [62] J. M. Weisberg and J. H. Taylor. *Relativistic Binary Pulsar B1913+16: Thirty Years of Observations and Analysis*. 2004. arXiv: [astro-ph/0407149](https://arxiv.org/abs/astro-ph/0407149) [astro-ph].
- [63] B. P. Abbott et al. “Observation of Gravitational Waves from a Binary Black Hole Merger”. In: *Phys. Rev. Lett.* 116 (6 2016), p. 061102. DOI: [10.1103/PhysRevLett.116.061102](https://doi.org/10.1103/PhysRevLett.116.061102). URL: <https://link.aps.org/doi/10.1103/PhysRevLett.116.061102>.
- [64] B. P. Abbott et al. “GWTC-1: A Gravitational-Wave Transient Catalog of Compact Binary Mergers Observed by LIGO and Virgo during the First and Second Observing Runs”. In: *Phys. Rev. X* 9.3 (2019), p. 031040. DOI: [10.1103/PhysRevX.9.031040](https://doi.org/10.1103/PhysRevX.9.031040). arXiv: [1811.12907](https://arxiv.org/abs/1811.12907) [astro-ph.HE].
- [65] B. P. Abbott et al. In: 848.2 (2017), p. L12. DOI: [10.3847/2041-8213/aa91c9](https://doi.org/10.3847/2041-8213/aa91c9). URL: <https://doi.org/10.3847/2041-8213/aa91c9>.
- [66] Rosalba Perna, Davide Lazzati, and Will Farr. “Limits on electromagnetic counterparts of gravitational wave-detected binary black hole mergers”. In: *Astrophys. J.* 875.1 (2019), p. 49. DOI: [10.3847/1538-4357/ab107b](https://doi.org/10.3847/1538-4357/ab107b). arXiv: [1901.04522](https://arxiv.org/abs/1901.04522) [astro-ph.HE].
- [67] Davide Miceli et al. “Following up GW Alerts With MAGIC: The Third LIGO/Virgo Observation Run”. In: *MAGIC Contributions to the 36th International Cosmic Ray Conference (ICRC2019)*. 2019. arXiv: [1909.03971](https://arxiv.org/abs/1909.03971) [astro-ph.HE].
- [68] Monica Seglar-Arroyo et al. “Searches for counterparts of gravitational waves at very high energies with H.E.S.S.”. In: (2019). arXiv: [1908.06705](https://arxiv.org/abs/1908.06705) [astro-ph.HE].

- [69] Marcos Santander. “Recent results from the VERITAS multimessenger program”. In: *PoS ICRC2019* (2019), p. 782. arXiv: 1909.05228 [astro-ph.HE].
- [70] David Shoemaker. “Gravitational wave astronomy with LIGO and similar detectors in the next decade”. In: (2019). arXiv: 1904.03187 [gr-qc].
- [71] Knödlseder, J. et al. “GammaLib and ctools - A software framework for the analysis of astronomical gamma-ray data”. In: *A&A* 593 (2016), A1. DOI: 10.1051/0004-6361/201628822. eprint: <http://ascl.net/1601.005>.
- [72] C. Deil et al. “Gammapy - A prototype for the CTA science tools”. In: *35th International Cosmic Ray Conference (ICRC2017)*. Vol. 301. International Cosmic Ray Conference. 2017, p. 766. arXiv: 1709.01751 [astro-ph.IM].
- [73] H. Abdalla et al. “The population of TeV pulsar wind nebulae in the H.E.S.S. Galactic Plane Survey”. In: *Astronomy & Astrophysics* 612 (2018), A2. ISSN: 1432-0746. DOI: 10.1051/0004-6361/201629377. URL: <http://dx.doi.org/10.1051/0004-6361/201629377>.
- [74] P. Cristofari et al. “Acceleration of cosmic rays and gamma-ray emission from supernova remnants in the Galaxy”. In: *Monthly Notices of the Royal Astronomical Society* 434.4 (2013), pp. 2748–2760. ISSN: 0035-8711. DOI: 10.1093/mnras/stt1096. URL: <https://doi.org/10.1093/mnras/stt1096>.
- [75] M. Ackermann et al. “The First Fermi-LAT Catalog of Sources Above 10 GeV”. In: *The Astrophysical Journal Supplement Series* 209.2 (2013), p. 34. ISSN: 1538-4365. DOI: 10.1088/0067-0049/209/2/34. URL: <http://dx.doi.org/10.1088/0067-0049/209/2/34>.
- [76] Alberto Franceschini, Giulia Rodighiero, and Mattia Vaccari. “The extragalactic optical-infrared background radiations, their time evolution and the cosmic photon-photon opacity”. In: *Astron. Astrophys.* 487 (2008), p. 837. DOI: 10.1051/0004-6361:200809691. arXiv: 0805.1841 [astro-ph].
- [77] M. Ackermann et al. “2FHL: The Second Catalog of Hard Fermi-LAT Sources”. In: *Astrophys. J. Suppl.* 222.1 (2016), p. 5. DOI: 10.3847/0067-0049/222/1/5. arXiv: 1508.04449 [astro-ph.HE].
- [78] F. D’Ammando et al. “The first γ -ray detection of the narrow-line Seyfert 1 FBQS J1644+2619”. In: *Monthly Notices of the Royal Astronomical Society* 452.1 (2015), pp. 520–524. ISSN: 0035-8711. DOI: 10.1093/mnras/stv1278. URL: <https://doi.org/10.1093/mnras/stv1278>.
- [79] M. Ackermann et al. “Fermi-LAT Observations of the Gamma-Ray Burst GRB 130427A”. In: *Science* 343.6166 (2013), pp. 42–47. ISSN: 1095-9203. DOI: 10.1126/science.1242353. URL: <http://dx.doi.org/10.1126/science.1242353>.

- [80] E. Bissaldi et al. “Prospects for Gamma-Ray Burst detection by the Cherenkov Telescope Array”. In: *European Physical Journal Web of Conferences*. Vol. 136. European Physical Journal Web of Conferences. 2017, p. 03019. DOI: [10.1051/epjconf/201713603019](https://doi.org/10.1051/epjconf/201713603019).
- [81] Maria Grazia Bernardini et al. *POSyTIVE – a GRB population study for the Cherenkov Telescope Array (ICRC-2019)*. 2019. arXiv: [1908.01544](https://arxiv.org/abs/1908.01544) [astro-ph.HE].
- [82] A. Domínguez et al. “Extragalactic background light inferred from AEGIS galaxy-SED-type fractions”. In: *Monthly Notices of the Royal Astronomical Society* 410.4 (2010), 2556–2578. ISSN: 0035-8711. DOI: [10.1111/j.1365-2966.2010.17631.x](https://doi.org/10.1111/j.1365-2966.2010.17631.x). URL: <http://dx.doi.org/10.1111/j.1365-2966.2010.17631.x>.
- [83] D. Berge, S. Funk, and J. Hinton. “Background modelling in very-high-energy γ -ray astronomy”. In: *Astronomy & Astrophysics* 466.3 (2007), pp. 1219–1229. ISSN: 1432-0746. DOI: [10.1051/0004-6361:20066674](https://doi.org/10.1051/0004-6361:20066674). URL: <http://dx.doi.org/10.1051/0004-6361:20066674>.
- [84] Monica Seglar-Arroyo et al. *The gravitational-wave follow-up program of the Cherenkov Telescope Array*. 2019. arXiv: [1908.08393](https://arxiv.org/abs/1908.08393) [astro-ph.HE].
- [85] *GW COSMoS: Gravitational Wave COmpact binary System Simulations*. DOI: [10.6084/m9.figshare.c.4243595.v1](https://doi.org/10.6084/m9.figshare.c.4243595.v1). URL: https://figshare.com/collections/GW_COSMoS_Gravitational_Wave_COmpact_binary_System_Simulations/4243595/1.
- [86] K. Kosack and M. Peresano. “ctapipe: A Low-level Data Processing Framework for CTA”. In: vol. ICRC2019. 2019. eprint: <https://pos.sissa.it/358/717/pdf>. URL: <http://github.com/cta-observatory/ctapipe>.
- [87] Travis E Oliphant. *A guide to NumPy*. Vol. 1. Trelgol Publishing USA, 2006.
- [88] Eric Jones, Travis Oliphant, Pearu Peterson, et al. *SciPy: Open source scientific tools for Python*. [Online; accessed]. 2001–. URL: <http://www.scipy.org/>.
- [89] Astropy Collaboration et al. “The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package”. In: *Astron. J.* 156.3, 123 (2018), p. 123. DOI: [10.3847/1538-3881/aabc4f](https://doi.org/10.3847/1538-3881/aabc4f). arXiv: [1801.02634](https://arxiv.org/abs/1801.02634) [astro-ph.IM].
- [90] John D Hunter. “Matplotlib: A 2D graphics environment”. In: *Computing in science & engineering* 9.3 (2007), pp. 90–95.
- [91] Wes McKinney et al. “Data structures for statistical computing in python”. In: *Proceedings of the 9th Python in Science Conference*. Vol. 445. Austin, TX. 2010, pp. 51–56.
- [92] Fabian Pedregosa et al. “Scikit-learn: Machine learning in Python”. In: *Journal of machine learning research* 12.Oct (2011), pp. 2825–2830.
- [93] Stefan Behnel et al. “Cython: The best of both worlds”. In: *Computing in Science & Engineering* 13.2 (2011), pp. 31–39.

- [94] Karl Kosack et al. *cta-observatory/ctapipe: v0.7.0*. Version v0.7.0. Aug. 2019. DOI: [10.5281/zenodo.3372211](https://doi.org/10.5281/zenodo.3372211). URL: <https://doi.org/10.5281/zenodo.3372211>.
- [95] S. Le Bohec et al. “A New analysis method for very high definition imaging atmospheric Cherenkov telescopes as applied to the CAT telescope”. In: *Nucl. Instrum. Meth. A* 416 (1998), pp. 425–437. DOI: [10.1016/S0168-9002\(98\)00750-5](https://doi.org/10.1016/S0168-9002(98)00750-5). arXiv: [astro-ph/9804133](https://arxiv.org/abs/astro-ph/9804133) [astro-ph].
- [96] Mathieu de Naurois and Loïc Rolland. “A high performance likelihood reconstruction of γ -rays for imaging atmospheric Cherenkov telescopes”. In: *Astroparticle Physics* 32.5 (2009), 231–252. ISSN: 0927-6505. DOI: [10.1016/j.astropartphys.2009.09.001](https://doi.org/10.1016/j.astropartphys.2009.09.001). URL: <http://dx.doi.org/10.1016/j.astropartphys.2009.09.001>.
- [97] M. Naumann-Godó, M. Lemoine-Goumard, and B. Degrange. “Upgrading and testing the 3D reconstruction of gamma-ray air showers as observed with an array of Imaging Atmospheric Cherenkov Telescopes”. In: *Astroparticle Physics* 31.6 (2009), pp. 421–430. ISSN: 0927-6505. DOI: <https://doi.org/10.1016/j.astropartphys.2009.04.006>. URL: <http://www.sciencedirect.com/science/article/pii/S0927650509000747>.
- [98] R. D. Parsons and J. A. Hinton. “A Monte Carlo Template based analysis for Air-Cherenkov Arrays”. In: *Astropart. Phys.* 56 (2014), pp. 26–34. DOI: [10.1016/j.astropartphys.2014.03.002](https://doi.org/10.1016/j.astropartphys.2014.03.002). arXiv: [1403.2993](https://arxiv.org/abs/1403.2993) [astro-ph.IM].
- [99] Mikaël Jacquemont et al. “GammaLearn: a Deep Learning framework for IACT data”. In: *36th International Cosmic Ray Conference*. Vol. ICRC2019. Madison, United States, 2019, p. 705. URL: <https://hal.archives-ouvertes.fr/hal-02197399>.
- [100] M. Szanecki et al. “Monte Carlo simulations of alternative sky observation modes with the Cherenkov Telescope Array”. In: *Astroparticle Physics* 67 (2015), pp. 33–46. ISSN: 0927-6505. DOI: <https://doi.org/10.1016/j.astropartphys.2015.01.008>. URL: <http://www.sciencedirect.com/science/article/pii/S0927650515000183>.
- [101] Cyril Trichard et al. “Enhanced H.E.S.S. II low energies performance thanks to the focus system”. In: *PoS ICRC2015* (2016). [34,1028(2015)], p. 1028. DOI: [10.22323/1.236.1028](https://doi.org/10.22323/1.236.1028). arXiv: [1509.02800](https://arxiv.org/abs/1509.02800) [astro-ph.HE].
- [102] G. Dubus et al. “Surveys with the Cherenkov Telescope Array”. In: *Astroparticle Physics* 43 (2013). Seeing the High-Energy Universe with the Cherenkov Telescope Array - The Science Explored with the CTA, pp. 317–330. ISSN: 0927-6505. DOI: <https://doi.org/10.1016/j.astropartphys.2012.05.020>. URL: <http://www.sciencedirect.com/science/article/pii/S0927650512001259>.

- [103] Lucie Gérard. “Divergent pointing with the Cherenkov Telescope Array for surveys and beyond”. In: *PoS ICRC2015* (2016), p. 725. DOI: [10.22323/1.236.0725](https://doi.org/10.22323/1.236.0725). arXiv: [1508.06197](https://arxiv.org/abs/1508.06197) [astro-ph.IM].
- [104] A. Donini et al. “The Cherenkov Telescope Array Performance in Divergent Mode”. In: *36th International Cosmic Ray Conference (ICRC2019)*. Vol. 36. International Cosmic Ray Conference, 2019, p. 664. arXiv: [1907.07978](https://arxiv.org/abs/1907.07978) [astro-ph.IM].
- [105] Kalevi Mattila and Petri Väisänen. “Extragalactic Background Light: Inventory of light throughout the cosmic history”. In: *Contemp. Phys.* 60.1 (2019), pp. 23–44. DOI: [10.1080/00107514.2019.1586130](https://doi.org/10.1080/00107514.2019.1586130). arXiv: [1905.08825](https://arxiv.org/abs/1905.08825) [astro-ph.GA].
- [106] V. A. Acciari et al. “Measurement of the extragalactic background light using MAGIC and Fermi-LAT gamma-ray observations of blazars up to $z = 1$ ”. In: *Mon. Not. Roy. Astron. Soc.* 486.3 (2019), pp. 4233–4251. DOI: [10.1093/mnras/stz943](https://doi.org/10.1093/mnras/stz943). arXiv: [1904.00134](https://arxiv.org/abs/1904.00134) [astro-ph.HE].
- [107] H. Martínez-Huerta et al. *Testing cosmology and fundamental physics with the Cherenkov Telescope Array*. 2019. arXiv: [1907.08141](https://arxiv.org/abs/1907.08141) [astro-ph.HE].
- [108] Mathieu de Naurois. “Very High Energy astronomy from H.E.S.S. to CTA. Opening of a new astronomical window on the non-thermal Universe”. habilitation. Ecole Polytechnique, 2012. URL: <http://tel.archives-ouvertes.fr/tel-00687872>.
- [109] G. Cavallo and M. J. Rees. “A qualitative study of cosmic fireballs and γ -ray bursts”. In: *Monthly Notices of the Royal Astronomical Society* 183.3 (1978), pp. 359–365. ISSN: 0035-8711. DOI: [10.1093/mnras/183.3.359](https://doi.org/10.1093/mnras/183.3.359). eprint: <http://oup.prod.sis.lan/mnras/article-pdf/183/3/359/2943441/mnras183-0359.pdf>. URL: <https://doi.org/10.1093/mnras/183.3.359>.
- [110] P. Kumar. “The Distribution of Burst Energy and Shock Parameters for Gamma-Ray Bursts”. In: *Astrophys. J. Lett.* 538 (Aug. 2000), pp. L125–L128. DOI: [10.1086/312821](https://doi.org/10.1086/312821). eprint: [astro-ph/9912566](https://arxiv.org/abs/astro-ph/9912566).
- [111] R. W. Klebesadel, I. B. Strong, and R. A. Olson. “Observations of Gamma-Ray Bursts of Cosmic Origin”. In: *Astrophys. J. Lett.* 182 (June 1973), p. L85. DOI: [10.1086/181225](https://doi.org/10.1086/181225).

Glossary

CTA Cherenkov Telescope Array.