Beamforming with AERA

Reconstruction Methods
for Radio Measurements
of Extensive Air Showers
at the Pierre Auger Observatory

Diplomarbeit
zur Erlangung des Akademischen Titels
Diplom-Physiker

dem Fachbereich Physik
der Bergischen Universität Wuppertal
vorgelegt von

— *Ingolf Jandt* —

WU D-11-16

Betreuender Professor:	Prof. Karl-Heinz Kampert
Zweitgutachter:	Prof. Klaus Helbing

To attain knowledge, add things every day;
to obtain wisdom, remove things every day.

Lao Tzu
Tao Te Ching

Contents

Abstract

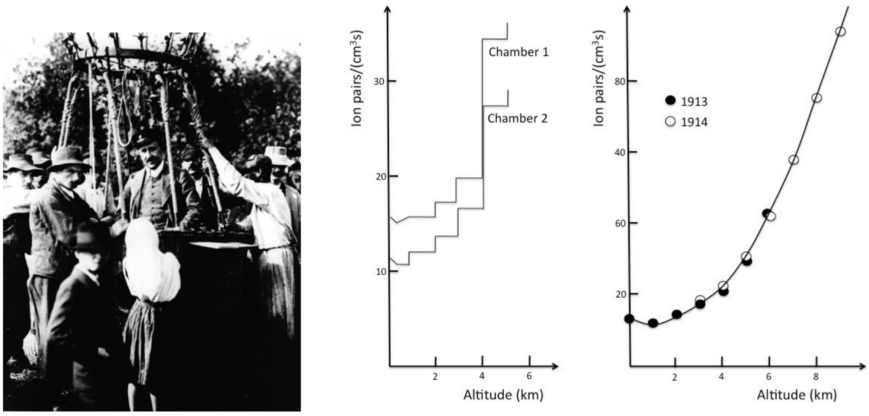
The Auger Engineering Radio Array is an array of 161 radio antennas for a frequency range of 30 to 80 MHz being setup at the Pierre Auger Observatory in Argentina. It records broad band radio pulses from Extensive Air Showers caused by cosmic ray particles. This study is to advance analysis technology by implementing beamforming methods into the analysis framework Offline.

Two different approaches are realized, both using the interferometric cross correlation to estimate the intensity emitted from a certain source position with its time dependance. The first method rasters these points and applies a wave model. It can create 2D and 3D visualisations. The other method optimizes the interference to improve the timing information that reflects the optical path lengths, thus improving accuracy of existing analysis methods.

Initial results show compatibility with coincident measurements from the surface detector. The rastering method shows deviations from the reference values similar to those of the standard reconstruction. This is explained by inaccuracies of the timestamps. Visualisations show a high sensitivity for the direction and limited one for the distance. A lobe-shaped structure can be seen in the images. The second method successfully improved direction reconstruction, limited by the timing inaccuracy.

1 Cosmic Rays

Cosmic rays are the incidence of high energy ionizing radiation from outer space onto the earth's atmosphere. In 1912 Victor Hess discovered them measuring the ionization at different heights with a Wulf electrometer. The expectation was a steady decrease with the distance from the ground with its natural radioactivity, the dominant source of known ionizing radiation. But as ionization grew again from 1000m on, Hess postulated a radiation from above, called *Höhenstrahlung* [1].



(a) Hess after balloon flight (b) Ionization data from Hess and Kolhörster, from [2]

Figure 1: Discovery of *Höhenstrahlung*

1.1 Chemical Composition

With direct measurements of cosmic rays (see 1.4) it is possible to identify the incoming particles. They indicate a composition of about 87% protons, 12% α particles and 1% heavier nuclei plus electrons and γ. ([3])

Figure 2 shows the elemental composition of cosmic rays up to 2 GeV/nucleon, compared to element occurrence in the solar system. The remarkable similarity may point to a common origin. The discrepancy at lithium, beryllium and boron ($Z = 3..5$) can be explained by spallation of oxygen and carbon during the propagation; the same goes for scandium ($Z = 21$) to manganese (25) as spallation products of iron.

Because the flux dramatically decreases with energy, only particles up to about 10^{15} eV can be measured directly. At higher energies, indirect measurements using air showers (section 1.4) must be interpreted by means of extrapolations and simulations. These yield uncertainties, so the composition at highest energies is largely unknown.

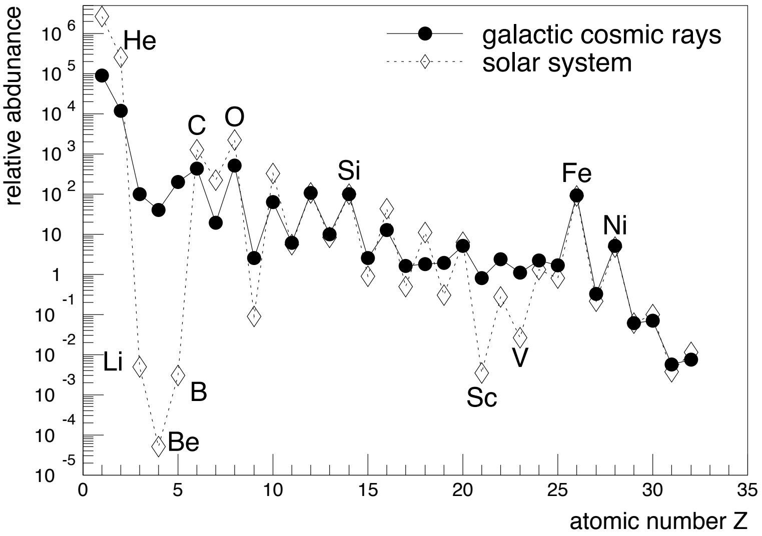


Figure 2: Chemical composition until 2 GeV/Z, normalized to Si [4]

1.2 Energy spectrum

In a large energy range the cosmic ray energy spectrum follows a power law

$$\frac{dN}{dE} \propto E^{-\gamma} \tag{1}$$

dropping fast, from one square meter getting one particle per second at around 10^{11} eV to one particle per year at $5 \cdot 10^{15}$ eV; above 10^{19} eV only one particle per century hits one square *kilometer*.

As figure 3 shows, the spectral index γ changes slightly at certain points. The spectrum was scaled with $E^{2.7}$ to make these features visible: γ changes from ~ 2.7 to ~ 3.1 at the *knee* at about $4 \cdot 10^{15}$ eV, then to ~ 3.3 at energies of about $5 \cdot 10^{17}$, the *second knee*, and flattens again to ~ 2.7 at the *ankle* at roughly $5 \cdot 10^{18}$ eV.

Above $5.7 \cdot 10^{19}$ eV the spectrum is cut off by the GZK effect (after Greisen [6], Zatsepin and Kuzmin [7]): The cosmic ray particles interact with the photons of the cosmic background radiation ($1.2 \cdot 10^{-3}$ eV), which in their rest frame becomes hard x-rays (i.e. 140 MeV $= m_\pi$). The reaction produces secondary particles, mainly pions via for example $\gamma + p \rightarrow \Delta^+ \rightarrow p + \pi^0$, reducing the kinetic energy of the primary particle. Therefore, a proton can retain an energy above the threshold for only a few 10 Mpc.

Measurements from AGASA [8] seemed to contradict this cut-off, suggesting ultra high energy sources relatively nearby in the local supercluster, not visible in other radiation. But later measurements by the Pierre Auger Observatory [9] and HiRES [10] did detect a suppression with high statistical significance.

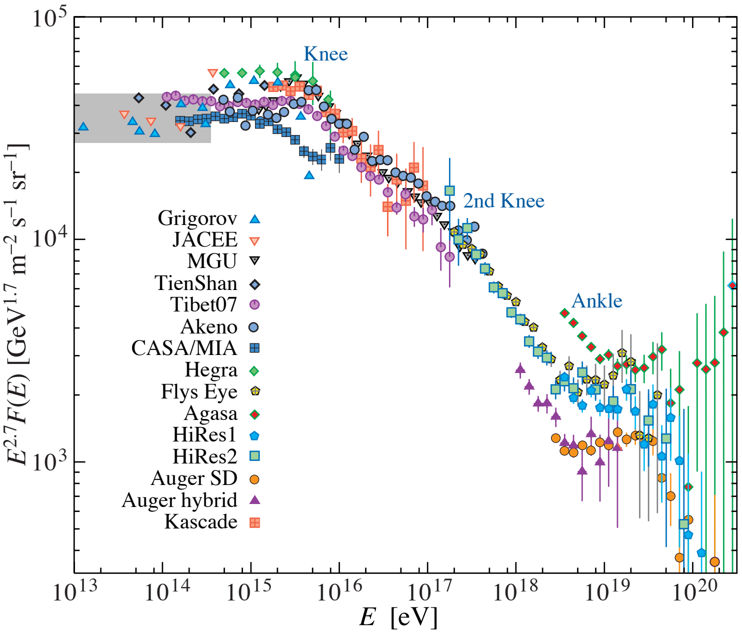


Figure 3: All particle energy spectrum, from [5]

1.3 Origin

The sources of high energy cosmic rays are so far uncertain. Given the large energy range and the spectrum features, probably different types of objects are responsible for the acceleration. Among the candidates are supernovae and jets of active galactic nuclei (AGN).

Usual acceleration models are first order and second order fermi acceleration ([11],[12]): A particle is deflected by the magnetic inhomogeneities around a shock front (i.e. from a supernova), crossing it repeatedly gaining energy of $\frac{\Delta E}{E} \propto \beta$ each time, where $\beta = \frac{v}{c}$ is the shock front speed. Or it is mirrored between two or more fronts, gaining $\frac{\Delta E}{E} \propto \beta^2$ if they are advancing[1] with β.

Both models result in exponential source spectra, as the probability to collect n portions of energy without escaping or losing energy in collisions diminishes with P^n, if P is the probability of a single successful step.

Apart from acceleration or bottom-up models, top-down scenarios were proposed to explain ultra high energy cosmic rays (UHECR) by the decay of heavy metastable particles in the nearby universe. Candidates are magnetic monopoles or topological defects left over from the big bang. But due to gravitation these should cumulate in the galactic disc or globular clusters. No according anisotropy is seen in the cosmic rays.

[1] This can also be inside a magnetic cloud with random movements, where head-on collisions equivalent to advancing mirrors are more probable

The milky way has an magnetic field of around 2 to 3 μG in the galactic disc. The corresponding gyro radius of particles up to 10^{15} eV is smaller than 1 pc, so there is no correlation of arrival direction and source location. Only from 10^{18} eV on, with a few hundred parsecs gyro radius, anisotropies are expected to point towards source regions.

1.4 Detection

Direct measurements According to [3], the typical mean free path of GeV protons in air is $\lambda = \frac{A}{N_A \cdot \sigma_A} \sim 90 \frac{g}{\mathrm{cm}^2}$, with nitrogen nuclei ($A = 14$) as the main interaction partners. Research balloons can reach heights of up to 40 km, which means that the cosmic rays cross about $5 \frac{g}{\mathrm{cm}^2}$ of rest atmosphere. So a relatively pure probe of primary particles can be taken. This holds even more for satellites, flying at hundreds of kilometers heights (low earth orbit, LEO: 200 to 1200 km).

With the dimensions of a balloon or satellite borne experiments and mission periods of months or years, cosmic rays above 10^{15} eV are too rare to be detected with a significant statistic.

Extensive air showers The atmospheric depth from space to sea level is about $1000 \frac{g}{\mathrm{cm}^2}$. So primary particles will not reach the earth. The proton–air cross section grows roughly logarithmic with energy, from $\sigma \sim 250$ mb at a few GeV (for the λ value above), to about 700 mb at 10^{18} eV, reducing the mean free path. So more energetic particles will react even higher in the atmosphere.

From the first collision with typically a nitrogen nucleus, a large number of secondary hadrons is produced, which in turn cause a cascade of further hadronic and electromagnetic reactions, partly decay and result in a front of millions of particles hitting the ground with nearly vacuum speed of light. This slightly curved front is a few meters thick and a few hundred meters wide, depending on primary energy.

The existence of such extensive air showers (EAS) was discovered 1938 by Auger [13], by coincidences of particle detectors mounted at 300 m distance in the swiss alps.

Figure 4 shows a schematic view of a shower. It consists of a hadronic component in the center with about 1 % of the shower energy, an electromagnetic component with 95 %, and a muonic component with 4 % of the energy.

By placing a grid of particle detectors – like Cherenkov detectors or scintillators – on the ground, the atmosphere itself can be used as a detector medium for the primary particles. Thus with appropriately large arrays, sufficient statistics can be attained even for the highest energy particles within years or decades.

The moving shower core can also be spotted by its fluorescence light: The passage of the charged particle cloud leaves excited N_2 and N_2^+ molecules. Upon de-excitation they emit

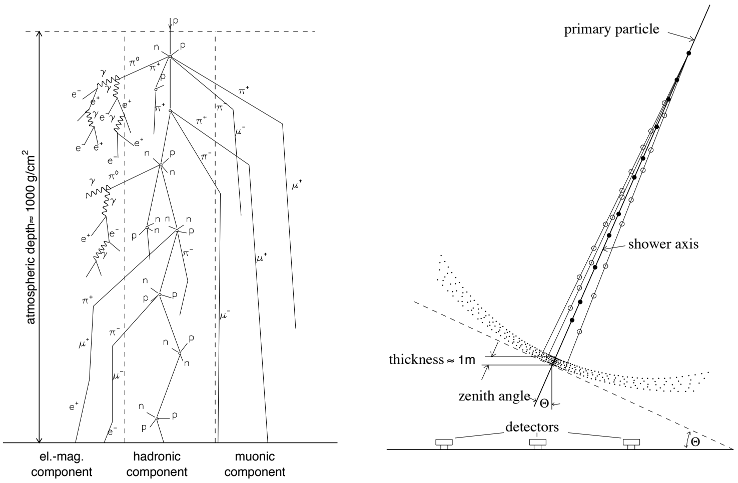


Figure 4: Extensive air shower, from [14]

radiation of $\lambda \approx 300..400\,\text{nm}$ from their 2P and 1N bands, isotropically. The whole emitted fluorescence makes up less than 1 % of the shower energy, so deep darkness is required for detection.

Radio emission Following theoretical predictions of Askaryan [15], in 1965 Jelly et al. detected radio emissions from EAS at 44 MHz [16]. Until 1970 a bandwidth from 2 to 520 MHz was found. [17]

Because of difficulties with radio frequency interferences (RFI), mainly caused by human noise, this way of detecting air showers was not much followed the years after. But when these issues are sorted out, this method has its advantages: As the radio signal sums up over the source region, it gives information on the whole shower evolution, complementary to ground based particle detectors. In contrast to fluorescence light measurements, radio signals do not suffer from clouds or moonlight, allowing nearly 100% duty time (thunderstorms cause radio interferences with huge field strengths). And they can be done with relatively cheap equipment: Broadband MHz radio antennas with adapted low noise amplifiers and DAQ electronic.

The earliest predictions by *Askaryan* describe radio emission as an effect of a negative *charge excess*: Whilst in pair production electrons and positrons are created in equal numbers, the latter annihilate with electrons of the air and do not contribute to the moving charge any more. So their effective lifetime at typical shower energies is only one tenth of the electron lifetime given by the radiation length. Together with Compton and δ electrons this leads to a buildup of not compensated negative charge. The additional electrons coherently add up their radiation. Askaryan concentrated on Cherenkovlike radiation, but recently [20] the time variation of the charge excess itself was shown to be a dominant contribution to the charge excess effect.

Experimental data showed a dependency of the radiation with the angle of the shower towards the earth's magnetic field ([17]). This is explained by two equivalent geomagnetic emission models: According to Kahn [18], the opposite deflections of negative and positive particles cause a current in the shower, which emits a dipole radiation. The microscopic model of Falcke and Gorham [19] regard the same emission as a superposition of synchrotron radiation from the deflected particles.

Recent simulations [21] and measurements [22] suggest that, whilst the geomagnetic effect is dominant, the charge excess (Askaryan) effect gives a significant contribution, growing with higher frequencies. The latter is supposed to concentrate in a ring of about 100 m radius and is independent of the arrival direction, the geomagnetic radiation varies with the angle towards the geomagnetic field.

2 The Pierre Auger Observatory

The Pierre Auger Observatory (PAO) is the largest detector array for cosmic rays of the world, spanning 3000 km^2. It is sited in the Pampa Amarilla (yellow prairie) in Mendoza, western Argentina 1400 m above sea level. Figure 5 shows the site layout, with the array of the surface detector (SD) overlooked from four fluorescence telescope stations (fluorescence detector, FD) around it, and auxiliary facilities.

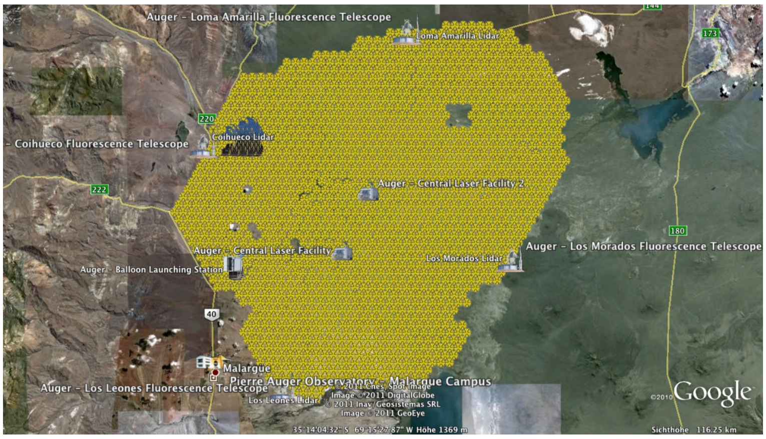

Auger - Loma Amarilla Fluorescence Telescope
Loma Amarilla Lidar
144
173
220
Coihueco Lidar
- Coihueco Fluorescence Telescope
222
Auger - Central Laser Facility 2
180
Auger - Los Morados Fluorescence Telescope
Los Morados Lidar
Auger - Central Laser Facility
Auger - Balloon Launching Station
40
Malargüe
Pierre Auger Observatory - Malargue Campus
Auger - Los Leones Fluorescence Telescope
Los Leones Lidar
© 2011 Cnes/Spot Image
Image © 2011 DigitalGlobe
© 2011 Inav/Geosistemas SRL
Image © 2011 GeoEye
35°14'04.32" S 69°15'27.87" W Höhe 1369 m
©2010 Google
Sichthöhe 116.25 km

Figure 5: PAO layout as seen in google earth, [23]

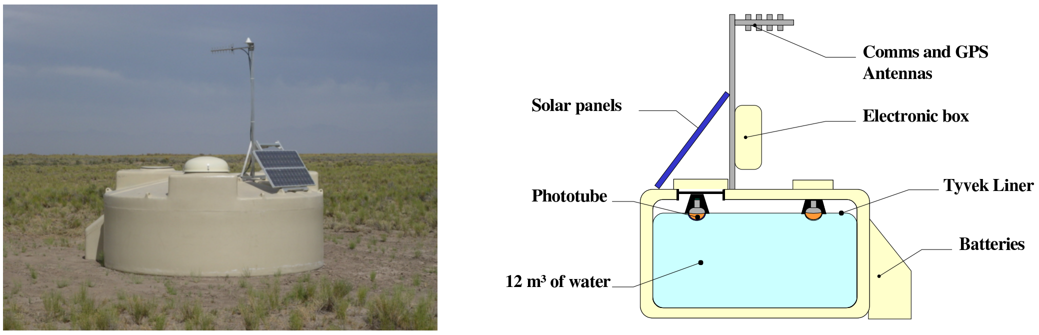
This hybrid design is, apart from its bare size, the most outstanding feature of the setup, measuring shower evolution in the atmosphere and the lateral distribution on the ground simultaneously. As a third component, the Auger Engineering Radio Array (AERA) is being built to study the radio emissions from EAS.

The status of the Pierre Auger Observatory is summarized in [24]. Here is a short overview of the different detectors co-located with AERA.

2.1 The Surface Detector (SD)

The surface detector consists of an array of 1600 cherenkov detector stations, placed on a triangular grid of 1500 m distance. Each station consists of a cylindrical tank with 10 m^2 base area and 1.2 m height, filled with 12 t of purified water. Three 9″ photomultipliers look downward into the water to observe Cherenkov light pulses.

AMIGA With the original spacing, the surface detector is fully efficient from $3 \cdot 10^{18}$ eV. To lower the detection threshold to 10^{17} eV for studying the region from second knee to ankle, the Auger Muons and Infill for the Ground Array (AMIGA) was added with two station spacings of 433 and 750m to account for the smaller showers at these energies. This infill



(a) SD station [25]

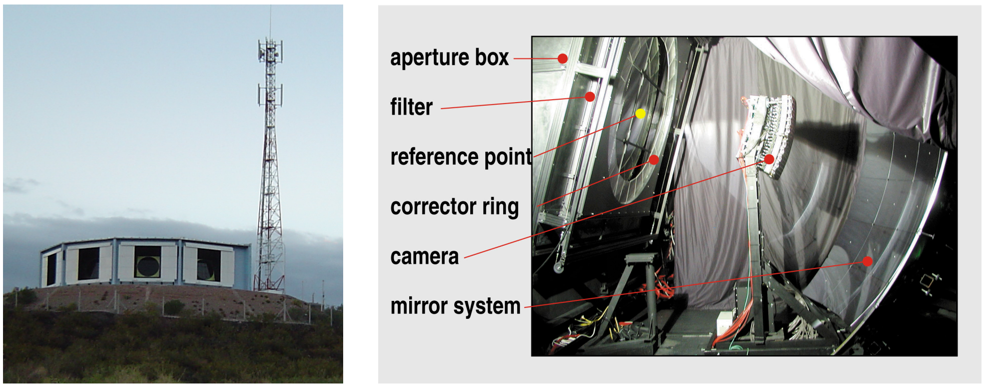
(b) SD water tank [26]

Figure 6: The surface detector stations are Cherenkov detectors: Cylindrical water tanks with three PMTs each

will consist of 85 SD stations accompanied by muon counters. It will cover an area of only 23.5 km^2, as the flux is much higher at these energies, and be situated near the Coihueco FD building.

2.2 The Fluorescence Detector (FD)

To measure the fluorescence light emitted by air showers, the four stations—named the eyes—of the fluorescence detectors (FD) overlook the SD array. Each one houses 6 Schmidt telescopes (figure 7b) with a 12 m^2 spherical mirror and a camera of 440 photomultipliers.



(a) Los Leones building [25]

(b) Fluorescence Telescope [25]

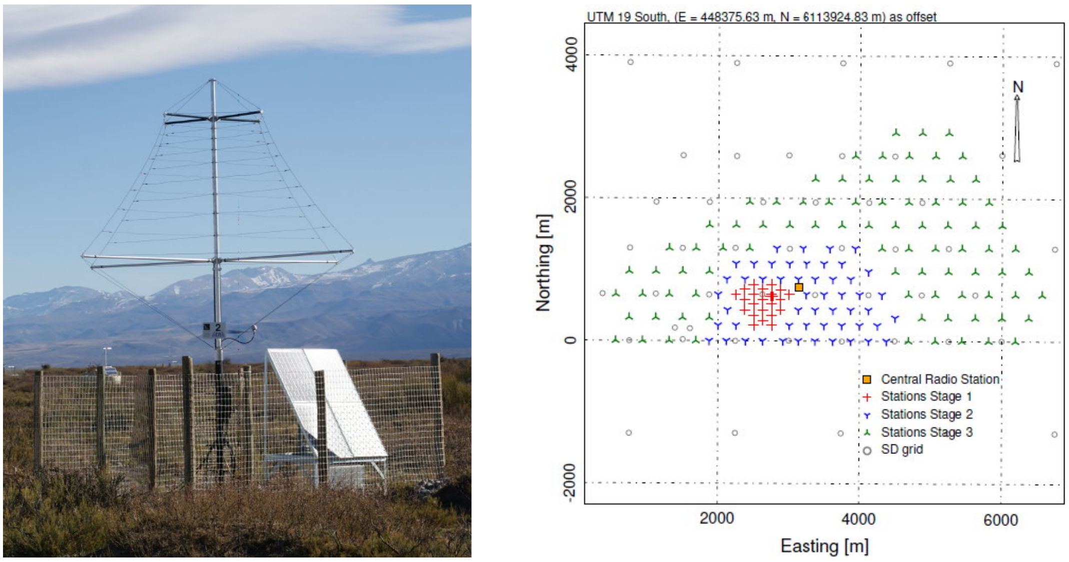
Figure 7: Six Schmidt cameras per “eye” constitute the fluorescence detector.

HEAT The *High Elevation Auger Telescopes* are three additional fluorescence telescope of similar design, with the ability to be tilted by 29° upwards, to a vertical field of view from 30° to 58.6°. This allows to detect showers developing higher in the atmosphere. This lowers the energy threshold to about 10^{17} eV.

2.3 AERA

In the last few years, experiments like Lopes [27] and Codalema [28] have shown that with today's technology arrays of radio detectors are a promising instrument to study cosmic rays of beyond 10^{17} eV. The *Auger Engineering Radio Array* (AERA) [29] is such an experiment. It will span 20 km^2 in the final stage, with three spacings in a trigonal grid: 24 stations at 150 m spacing, 52 stations with 20 m and finally 85 stations with 375 m spacing.

The different spacings reflect the different energies and their fluxes: The core is fully effective from $10^{17.5}$ eV on; there the remaining grid only contributes by 10% effective area, but the flux is high. 75% effective area for the whole array is reached at $10^{19.5}$ eV.



(a) AERA station: Two crossed LPDA with custom electronic [30]

(b) AERA stages layout near Coihueco (zero coordinates) [30]

Figure 8: The Auger Engineering Radio Array is a radio detector with 161(+2) stations spanning an area of 20 km^2

The stations (figure 8a) consist of two crossed *logarithmic periodic dipole antenna* (LPDA), aligned with the magnetic north [31], with low noise amplifiers (LNA) [32] and DAQ electronic. LPDAs combine a broad bandwidth with high directional gain, in this case towards the zenith. Antennas and amplifiers are optimized for the design bandwidth of 30 to 80 MHz.

Timing is synchronized via GPS, which is to be improved with the help of a radio beacon (see section 3.3).

Figure 8b shows the layout a few kilometers east of Coihueco (zero point), colocated with the Amiga infill and overlooked by HEAT. AERA is expected to record about 5000 events per year with a detection threshold of $10^{17.2}$ eV. This layout gives the opportunity for studies with hybrid or superhybrid events, measured by two or all detectors: In the first place shower shower development and emission mechanisms; the capabilities of radio detection. In further steps to get a better insight into the spectrum and composition at the region from second knee to ankle between $10^{17.4}$ and $10^{18.7}$ eV, where the transition between galactic and intergalactic component is expected.

3 Beamforming

This section introduces the idea of beam forming as a interferometry method to obtain a picture of where electromagnetic radiation is coming from. The general method and the quantities are taken from [33], but with substantial changes.

Beam forming is a technique of signal processing, where an array of sensors is used for directionally selective receiving[2], taking advantage from constructive and destructive interference. The incoming signals are combined such that for an expected direction a high output signal is seen. To achieve this, the individual signals are shifted according to the expected delay. That is, when a signal should arrive first at station A, and after a time delay τ at station B, the signal of B must be shifted by $-\tau$, backward in time, before the signals are interfered; this happens only numerically in the computer. With the correct shift applied, the signals will match best and give the highest output of the beam forming quantities described below.

How to calculate signal arrival delays dependent on the wave model is described in section 5.4, to keep the discussion of the beam forming quantities general. When the possible directions (and their respective sets of delays) are stepped through a grid, a two- or three-dimensional image can be made with the beam forming quantities serving as time dependent intensity measures for each source position. By scanning through these parameters, the time and source spot of maximum intensity can be found.

3.1 Beamforming Quantities – Motivation and Formulas

If one regards the superposition of two arbitrary electromagnetic waves $V_i(\vec{r}, t)$, the measured intensity at one "observer" point in space is

$$\begin{aligned} I(\vec{r},t) &= \left\langle |V_1(\vec{r},t) + V_2(\vec{r},t)|^2 \right\rangle = \langle (V_1(\vec{r},t) + V_2(\vec{r},t))(V_1^*(\vec{r},t) + V_2^*(\vec{r},t)) \rangle & (2) \\ &= \left\langle |V_1(\vec{r},t)|^2 \right\rangle + \left\langle |V_2(\vec{r},t)|^2 \right\rangle + 2\,\mathrm{Re}\langle V(\vec{r},t)V^*(\vec{r},t)\rangle & (3) \\ &= \underbrace{I_1(\vec{r},t) + I_2(\vec{r},t)}_{\mathcal{A}} + \underbrace{2\,\mathrm{Re}\,\langle V_1(\vec{r},t)V_2^*(\vec{r},t)\rangle}_{\mathcal{B}} & (4) \end{aligned}$$

When the electric fields are uncorrelated, the mutual product in $\mathcal{B}$ will disappear, and the sum of the intensities $\mathcal{A}$ is measured. So $\mathcal{A}$ contains the uncorrelated intensities, whilst $\mathcal{B}$ is a measure of the signal correlation.

For beam forming, the model is turned around: We interfere one wave observed at different times t_i and observer places $\vec{r}_i$, then $V_i(\vec{r},t) = V(\vec{r}_i, t_i)$. The mean value in $\mathcal{B}$ becomes

$$\Gamma(\vec{r}_1, \vec{r}_2, t_1, t_2) := \langle V(\vec{r}_1, t_1) \cdot V^*(\vec{r}_2, t_2) \rangle \qquad (5)$$

and is called the *mutual intensity* in interferometry. This is the base of what we call *cross correlation beam*, part $\mathcal{A}$ leads to the *power beam* described below.

For a deeper insight into the basics of interferometry, see [34].

[2] A complementary method can be used for directional transmission.

Power beam This quantity sums up the intensities measured at the N antenna station positions, normalized by N, giving in effect the mean intensity.

$$p(t) = \frac{1}{N}\sum_{i=1}^{N} |\vec{E}_i(t)|^2 \tag{6}$$

It is high when much power is coming from the antennas, regardless of the coherence of the signal.

When applying equation 6 to get intensity measures for different source position in the sky, the electrical fields will already be shifted by the corresponding signal arrival delays τ_i according to a certain wave model: $\vec{E}_i(t) = \vec{E}_{i,\text{measured}}(t_i + \tau_i)$.

Cross correlation beam The cross correlation sums up the dot products of the electrical fields from two stations at a time.

$$cc(t) = \frac{1}{M} \cdot \sum_{i=1}^{N}\sum_{j>i}^{N} \vec{E}_i(t) \cdot \vec{E}_j(t) \qquad\qquad M = \frac{N(N-1)}{2} \tag{7}$$

In effect this is linear to the sum of mutual intensities of all pairs of stations, normalized to the number M of pairs. As in the general case of interferometry as described above, this is a measure of how coherent the $\vec{E}$ field signals are; if one assumes that maximum overall coherence is given when two stations at a time fit best.

Also in this formula the shift $\vec{E}_i(t) = \vec{E}_{i,\text{measured}}(t + \tau_i)$ is presupposed.

The name *cross correlation* comes from signal processing, where the quantity

$$(f \star g)(t) \overset{\text{def}}{=} \int_{-\infty}^{\infty} f^*(\tau)\, g(t+\tau)\, d\tau \tag{8}$$

gives a measure for the similarity of the functions f and g dependent on the shift t. Although this equation has a similar form and purpose, note that here the functions are integrated to one value per time shift. The formula can be read as a convolution or a time shifted scalar product (in function space). Our cross correlation sums up the $\vec{E}$ field products for each time separately, leading to a time dependent quantity more like the mutual intensities Γ described above.

Note that [33] introduces an additional "signed square root" to the *power* and *cross correlation* formulae given here to make them linear in the signal amplitude. In this work I adhere to the usual definitions which have the physical meaning of intensities. Also note that the software framework delivers and handles full 3D $\vec{E}(t)$ vectors, so translation of the shown formulae into code is straight forward, and no consideration of vector components (and the rotation of the local coordinate system) needs to be done here.

Excess Beam Lopes [33] also introduces a third quantity, the so-called *excess beam x*, that normalizes the cross correlation beam to the power beam.

$$x(t) = cc(t) \cdot \left| \frac{cc(t) - \overline{cc}}{p(t) - \bar{p}} \right| \tag{9}$$

This has its roots in the *normalized cross correlation* applied mostly in image processing when comparing two pictures shifted by u, v

$$\gamma(u, v) = \frac{\sum_{x,y} \left[f(x,y) - \bar{f}\right] [g(x-u, y-v) - \bar{g}]}{\sqrt{\sum_{x,y} [f(x,y) - \bar{f}]^2 \sum_{x,y} [g(x-u, y-v) - \bar{g}]^2}} \tag{10}$$

to correct for bright regions without real correlation. Here again the sum is taken over all variables, whilst x is left t dependent.

Whilst normalization to the power is not a bad idea, the given formula has the problem to diverge for small p values. In Lopes, this is solved via an ad-hoc lower bound for the denominator $p(t) - \bar{p}$. I do not see this as an optimal or well-understood choice. In this implementation, the beam forming quantities are always normalized to their respective RMS when being compared. This should have the side-effect of suppressing noise sufficiently. The excess trace is calculated as given here, so it can be studied further.

3.2 Advantages

An obvious advantage of beamforming is the ability to create images of the intensity measures. Visualizing the radio emitting part of the shower, including timing information, gives a unique opportunity to study the shower evolution based on radio measurements.

Also quantifying properties of the shower, and thus finally of the primary particles, benefits from beamforming. Because the interferometric method uses both the amplitude and the phase information contained in the radio signal, its signal-to-noise ratio (SNR) improves with approximately the number of stations N. In contrast, the quality of a fit based on the pulse arrival time improves only with the statistical power proportional to $\sqrt{N}$.

It was shown at Lopes that the cross correlation is a tool to measure the arrival direction [35], the energy of the primary particle via the intensity [36], and its mass via $X_{\max}$ [37] .

3.3 Preconditions

As beam forming uses the phase information of coherent waves, timing must be known very precisely: The bandwidth of AERA from 30 to 80 MHz corresponds to periods between 12.5

to 33.3 ns. For beamforming, the timing precision should be at least one order of magnitude higher, about 1 ns. For reference, the current AERA data aquisition hardware digitizes with a sampling rate of 200 MHz, corresponding to a 5 ns time resolution. Tests [33] at Lopes, wich has a bandwidth of 43MHz to 76MHz, have shown that introducing an offset of 1.25 ns in a two-station beam already visibly reduces coherence, lowering the peak.

To get the relative timing to precisions even below 1 ns, a radio beacon was installed at the FD telescope Coihueco, and a system to correct the GPS-based timestamps is being added to the analysis software [38]: The beacon sends constant sine waves at four frequencies. With the relative phase information of this signal recorded by the AERA stations, the difference between arrival time and station clock can be measured. Because the beacon uses four frequencies, this works for deviations of a few periods. The method is explained in detail in [39].

So far, the timing only depends on the GPS signal. First studies have shown that the timing not only shows a variation from event to event by 2.5 ns, there is also a day-night drift in the *relative* timing of about 20 ns. Measurements with an Octocopter sending radio pulses at the AERA site have shown additional station timing offsets in the region of 20 ns. [40]

4 The Offline software

This section gives a short overview over the Offline software framework. Offline is a tool to analyze air shower data, used by all PAO experiments. Input data can be measurements of the detectors ("real events") or results from various simulation programs. Output are reconstructed shower properties like energy and direction. For a detailed description see [41].

4.1 Structure

Offline is a modular C++ software system. Its components fall roughly into the following categories:

Detector: The detector description provides a gateway for reading detector properties like station positions, antenna response patters etc.

Event: The event structure accumulates measured, simulated and reconstructed event data. Filling and reading the event structure is the way modules relay data to one another.

Modules: The work is performed by the modules, which encapsule the algorithms for reading in, analyzing and writing out event data. A module performs one clearly delimited step and usually relies on data written into the event structure by earlier modules.

Utilities: This category covers mathematical and physical models and tools like coordinate systems, fourier transformations to be used by the modules, as well as service utilities like data managers and run controllers. Latter provide the link between the other components.

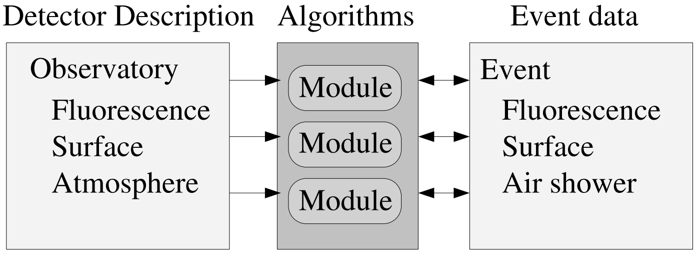


Figure 9: Structure of the Offline Framework, from [41]

In-memory representation of detector and event data are decoupled from file (or database) representation. So neither user nor module programmers have to care about the file formats from different detector stages or simulation packages.

4.2 Radio addition to the structure

Capabilities to handle radio signals and electric fields were added to Offline as described in [42], containing AERA detector description, modules to analyse radio data, and the `REvent` data structure for reconstructed properties. The latter comprises station related data, in the `RRecStation` substructure, and quantities of the shower in `RRecShower`.

Specific features used for beamforming are introduced along the way, without going into technical subtleties.

4.3 Usage

Being a software framework Offline provides the infrastructure for creating a vast variety of custom analysis programs. This can be done purely with XML files. The user has to provide at least two files: A file idiomatically called `ModuleSequence.xml` listing the modules to run, including control structure. This module sequence forms the *analysis pipeline*. The second file is `bootstrap.xml` which links to the XML files to be loaded.

A typical application will contain further XML files to overwrite default parameters of some of the modules.

4.4 Writing Modules

Building an analysis pipeline with provided modules can be done without C++ programming by chaining modules as mentioned above. But to add new analysis functionality, own modules had to be written as outlined here.

The core of the module is a C++ class inheriting from `fwk::VModule`. It must provide the methods `Init`, `Run` and `Finish`, each returning a status enum of the type `fwk::VModule::ResultFlag`. The module controller will make instances of all modules mentioned in the module sequence file and execute their `Init` method on startup.

Then it will `Run` them following the sequence, which usually implies looping through events and iterative calculations. With the return flag, `Run` can not only signal success but also request a loop to break or skip to the next iteration. Because of this partition of the control structure, modules must be designed carefully with the possible looping in mind; and can be used properly only in a matching sequence.

Finally, `Finish` will be called for every module, allowing cleanup of memory, closing files, writing out statistics and such.

Usually a module will have configuration options. In this case, an XML file with default settings and an XSD for validation are to be provided. The options are parsed by the configuration manager, and modules take over the values in their `Init` methods.

5 Implementation

This section describes the steps to raster the sky and do beamforming for every raster point and their implementation. It makes heavy use of Offline's capabilities [41], [42] to handle physical quantities.

The procedural approach was inspired by the Lopes software [43], [44]. But as the Offline system strongly differs in capabilities and underlying design decisions, the method had to be reimplemented completely from scratch.

5.1 Procedure

Beam forming consists of choosing a direction in the sky, calculating the corresponding signal arrival delays, shifting the recorded signals according to these delays and combining them according to the interferometry formulae.

This study uses the vectorial electrical field as input signal, so the vectorial $\vec{E}$ field traces have to be reconstructed. Because the RdAntennaChannelToStationConverter module (section 5.3) needs an input direction to guess the third component of $\vec{E}(t)$, the mentioned steps must be separated into two modules: One to preset the direction, another doing the calculation, with the RdAntennaChannelToStationConverter module running inbetween.

Listing 1 shows a module sequence as used by the provided RdBeamForming example application. An outer loop reads in events and prepares them as mentioned. Then in an inner loop RdPolarGrid rasters the sky in φ and ϑ incoming directions. RdAntennaChannelTo-StationConverter reconstructs the $\vec{E}$ signal, RdBeamFormer performs the interferometry calculations.

The modules earlier in the sequence deal with preparing the channel signal, gradually eliminating the influence of the detector chain characteristic. Events with less then 3 or 4 stations are sorted out with the RdEventPreSelector module. The traces are upsampled with RdChannelUpsampler with a factor of at least 4 to obtain circa nanosecond resolution, because the cross correlation oscillates fast in time and the result traces keep the input's time binning.

The module sequence is analog to the standard reconstruction, only the inner loop is replaced: The standard pipeline iteratively applies a plane wavefront fit to the pulse arrival times.

5.2 Grid Module

The grid module is a very lightweight module that serves two main purposes: Presetting a direction in a raster for the further sequence and breaking the loop when the whole sky

Listing 1: An example `ModuleSequence.xml.in` for measured data

```
<!-- Example sequence for radio beam forming with AERA data. -->

<sequenceFile xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
    xsi:noNamespaceSchemaLocation=’@SCHEMALOCATION@/ModuleSequence.xsd’>

  <moduleControl>
    <loop numTimes=”unbounded”>
      <module> EventFileReaderOG </module>
      <module> RdEventPreSelector </module>

      <module> RdChannelADCToVoltageConverter </module>
      <module> RdChannelPedestalRemover </module>
      <module> RdChannelResponseIncorporator </module>
      <module> RdChannelUpsampler </module>
      <module> RdChannelBandpassFilter</module>

      <loop numTimes=”unbounded”>
        <module> RdPolarGrid </module>
        <module> RdAntennaChannelToStationConverter </module>
        <module> RdBeamFormer </module>
      </loop>
      <module> RecDataWriterNG </module>
    </loop>
  </moduleControl>

</sequenceFile>
```

region is scanned. It does so very straightforward by incrementing one integer counter per coordinate and transforming it according to this coordinates' boundaries from the XML config file. When the highest order counter reaches its maximum, the run method breaks the loop by returning `eBreakLoop`.

The resulting coordinates are assembled to a vector in the chosen coordinate system (see below) and filled into the `axis` field of the radio related shower reconstruction data. This is the place where the `RdAntennaChannelToStationConverter` module expects its input direction. This choice has the advantage that no new data field is required to communicate the current direction, but the drawback that a potential earlier reconstruction result (i.e. a plane wave reconstruction with `RdPlaneFit`) is overwritten. So a previous value must be stored on entering the grid and restored when leaving it.

As the grid module is the earlier one in the pipeline, it also has some initialization tasks. On the physics side, it prepares the coordinate system to be used: Every offline vector (data type `utl::Vector`) holds a reference to its coordinate system and abstracts away a complex machinery to transform between local cartesian or polar coordinate systems on the curved earth. The grid module selects the reconstructed core position from a preceding analysis, if any, and has the framework calculate the local coordinate system for this origin, with the corresponding zenith, northing and easting.

If no reconstructed shower quantities exist, it is substituted with the unweighted mean position of the stations present in the event. This is a quick estimate of a center position: With the irregular and large AERA extent, the curved earth, and the off-center finer stages, there is no straight forward choice of a fixed, event independent center; on the other hand a more elaborate estimate like the barycenter requires a complete recursive reconstruction: The relative electric field strength is reconstructed dependent on the direction reconstruction. The same is done for the event time: If a preceding reconstruction is missing, the mean trace start times are used as a core time estimate, since the beam forming needs a constant reference time.

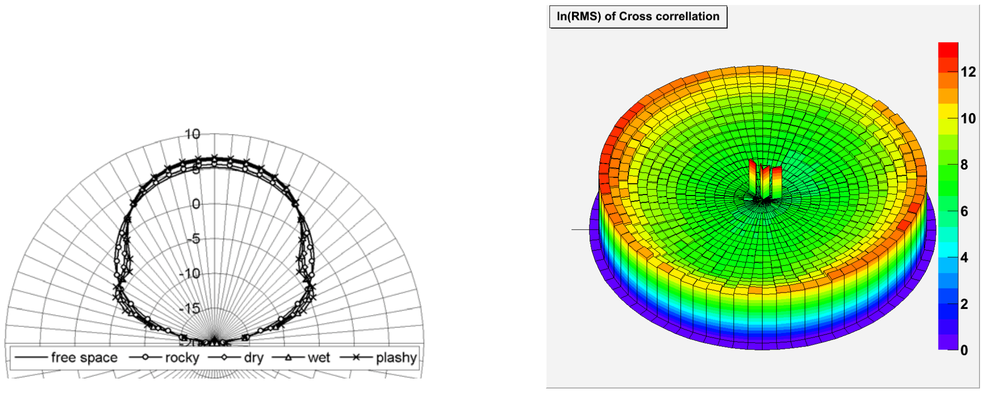
The other initialization concerns the output data fields: The grid module ensures the existence of a `BeamMap` object to hold the complete beam forming results and provides it with meta information of the type `BeamGrid` about the grid's stepping and boundaries. For more on the output data refer to section 5.5.

At the time of this writing, there are separate models for polar and cartesian grids. They are however being merged. As the former axis is already saved, a simple extension will be to scan relative to a former result, mainly the configure options must be added. Also an iterative grid refinement can be added here, by replacing the code breaking the loop by a new initialization, decrementing a step counter. This must be carefully designed however: A dynamic grid adjustment with a static configuration pattern, enough overlap to avoid local maxima, without overgeneralizing the problem.

5.3 E Field Reconstruction

The electric field at the station is reconstructed by the module `RdAntennaChannelToStationConverter`. It takes into account the direction-dependent sensitivity of the antennas and reconstructs the three-dimensional $\vec{E}$ field. Because by the two antennas a two-dimensional projection is measured, the third component must be guessed based on the transverse nature of electromagnetic waves.

Mathematically the channel signals are dot products of the incoming field with the direction-dependent effective antenna height. The operation of this module is inverting these equations with the additional constraint that $\vec{E} \cdot \vec{r}$ cancels.



(a) Antenna response pattern in dBi from [45] (LPDA, 2.5m above ground, 60 MHz)

(b) RMS of cross correlation trace in $\ln\left((\mu\mathrm{V}/\mathrm{m})^2\right)$

Figure 10: The varying antenna sensitivity biases intensity estimation towards horizon.

This module was written by Benjamin Fuchs and only slightly improved by me. It is described here being an crucial ingredient of beam forming with complete 3-dimensional $\vec{E}$ field, which is an outstanding property of the method showcased here.

It must be kept in mind, that applying a varying antenna sensitivity to the same channel signal results in a electric field estimate changing by orders of magnitude: Where the antenna gain gets minimal, the algorithm assumes a large electric field to reach the detected signal strength. Figure 10 demonstrates this effect. The directional diagram at the left shows the logarithmic antenna gain dependent on the zenith angle as simulated in [45]. The right diagram shows the RMS of an example event in sky map of zenith and azimuth angle.

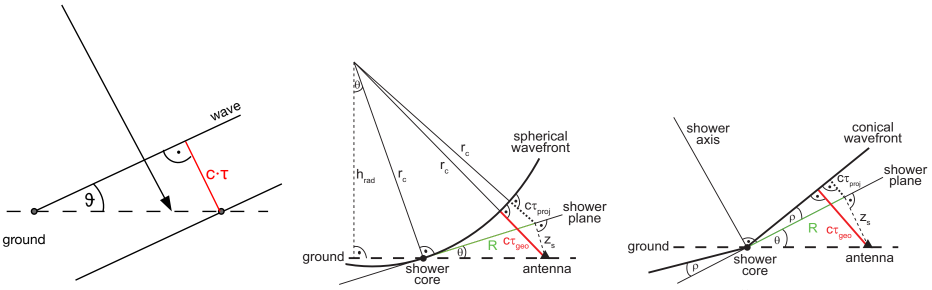
Bottom line: The absolute field strength of the result trace is not a benchmark of the maximum position, but rather reflects the antenna response. It must be normalized by its RMS.

5.4 Beamforming Module

Wave Models Calculation of the geometrical signal delays is kept separate from the calculation of beam quantities by a set of wave model classes: The base class `RdWaveModel` and the models `RdPlaneWaveModel`[3], `RdSphericalWaveModel` and `RdConicalWaveModel`. The respective delay methods are virtual, so the selected method is chosen at runtime without cluttering the module code with conditional statements. The module object contains an `RdWaveModel` pointer, the object it points to is created at the module's `Init` time, from the subclass corresponding to the wave model selected in the configuration.

With this scheme, the actual calculation code does not need to know about the wave model being used, not even what wave models are available. Only the `Init` method must allocate the chosen kind of object. This makes maintenance and extension of the code very easy.

Before requesting the delay, the wave model must be supplied with the coordinate system origin – the estimated shower core – and the current source position in the sky, both of type `utl::Point`. The available wave models depend only on the difference vector of these position (type `utl::Vector`). But it was handy during the design and debugging to use absolute positions; it also makes the implementation more readable, in my opinion. Future wave models may make use of the additional information. The run-time effect of this decision is "superfluously" adding the origin in the trace shift code and subtracting it again in the `delay` methods. This should not be too costly.



(a) Plane wave front (b) Spherical wave front (c) Conical wave front

Figure 11: Wave front models

The delays relative to the arrival at the origin are calculated in the respective `delay` method by the formulae easily derived from the models depicted in figure 11: For a *plane wavefront* it is

$$\tau = -\frac{1}{c} \cdot \frac{\vec{\rho} \cdot \vec{x}}{|\vec{\rho}|} \tag{11}$$

with the station position vector $\vec{x} = \overrightarrow{\mathrm{OA}}$ and the arrival direction vector $\vec{\rho} = \overrightarrow{\mathrm{OS}}$. Here O is the origin, A the antenna and S the source position. This formula is obviously just the

[3] The adjective "plane" here refers to a wave with planar wave fronts, the set of points with the same phase. Not a plane wave in the usual sense.

projection of $\vec{x}$ to the direction of $\vec{\rho}$, converted to the signal propagation delay by $\frac{1}{c}$. The minus sign comes from the source vector $\overrightarrow{OS}$'s looking upwards.

For a *spherical wavefront* the delay is

$$\tau = \frac{1}{c} \cdot (|\vec{\rho} - \vec{x}| - |\vec{\rho}|) \tag{12}$$

with $\vec{\rho} - \vec{x} = \overrightarrow{SA}$ the source-antenna distance. So to get the path difference the curved wave model literally subtracts the distances source—origin and source—antenna.

The conical wave model with opening angle of $180° - 2\,\alpha$ lengthens the path difference of the plane wave model according to the additional angle α:

$$\tau = -\frac{1}{c} \cdot (x_\perp \, \sin\alpha + x_{||} \, \cos\alpha) \qquad x_\perp = \frac{|\vec{\rho} \times \vec{x}|}{|\vec{\rho}|} \qquad x_{||} = \frac{\vec{\rho} \cdot \vec{x}}{|\vec{\rho}|} \tag{13}$$

The vector handling of Offline allows to transform these equations directly into code, without juggling with coordinate values. Besides from being more readable and writable, the vectorial modeling is proof against vertical variations in the station positions, when $\vec{x}$ is not parallel to the coordinate system plane any more.

Note that any wave can be regarded as a superposition of spherical waves expanding from a set of point sources (according to the Huygens-Fresnel principle). So the spherical model finds the spacial extent of the source. I.e. a point (ideally) if the wave front is really spherical, some kind of line if it is conical (or in a similar way rotation invariant around the $\vec{\rho}$ axis).

In other words, the models ask different questions to the signal. The spherical wave model is the candidate for three-dimensional imaging of the shower, the conical (or analogue) model for inspecting parameters of the resulting wave that reaches the ground.

In practice also the picture of a point source will be lengthy, because the angle sensitivity is higher than the distance sensitivity: Varying the tilt changes the path difference stronger and more oriented than varying the (small) curvature. So the smearing will be stretched parallel to $\vec{\rho}$ direction.

The plane front model comes into place where only the arrival direction is of interest. It has the advantage of being robust against origin displacements. The choice of origin still affects the angles indirectly, as it determines the local coordinate system axes. But this effect is tiny over the current AERA baseline, and can be taken into account in a further analysis, as the core point is saved into the result file.

Shifting Traces The code to shift the electrical field traces according to the delays relies on the abilities of the offline framework to perform on-the-fly fourier transformations with an API focused on physical quantities. Higher level code can thus concentrate on aspects of the physical model. So here some words on the data container provided by radio offline (see also [42]).

In the radio event data at station level, `revt::Station`, the actual radio signal is stored in a `StationFFTDataContainer`, an instance of the template `FFTDataContainer`. This type holds one storage object for the time and one for the frequency domain data and keeps track of which representation is up to date. When code requests a non-constant reference to one of the representations, it is marked as changed. When later the other format is requested, a fourier transformation is performed in the background, so the data received by higher level code is always up-to-date. With this mechanism physics code can concentrate on analyzing the data, on the other hand transformation are only done when needed.

In the case of `StationFFTContainer` which is used here, both storages are derived from the `utl::Trace` template: The time series representation `StationTimeSeries` is a trace of 3 dimensional double tuples (`utl::Vector3D`), for the E field data, the frequency representation `StationFrequencySpectrum` accordingly a trace of 3 dimensional tuples of *complex* doubles, `utl::Vector3C`.

With these technical details sorted out, shifting the traces by a desired interval Δt is very straightforward. Regarding the signal as a superposition of plane waves, each one's phase must be shifted by Δt's amount of the period, $\Delta\varphi_i = 2\pi \cdot \frac{\Delta t}{T_i} = 2\pi \cdot \Delta t \cdot \nu_i$. So we only have to iterate over the frequency spectrum and replace every value by

$$\tilde{\vec{E}}_{\text{shifted}}(\nu_i) = \tilde{\vec{E}}(\nu_i) \cdot \exp\left(-\imath \cdot 2\pi \cdot \Delta t \cdot \nu_i\right) \tag{14}$$

Note that $\exp(\imath\varphi)$ is shifted rightward (towards *positive* values) by $\exp\left(\imath(\varphi - \Delta\varphi)\right)$, hence the minus sign. Equation 14 is applied for every frequency bin in the `FFTDataContainerAlgorithm::shiftTimeSeries` method written by Frank Schröder. The `StationFFTDataContainer` takes care to deliver an updated time series at the next request, so nothing else needs to be done.

Before calling this method on copies of the station traces, some preparations are done. The code in the module method for shifting the traces deals with converting the computed delays to sensible trace shifts and presetting parameters for the subsequent interferometry. Most important, the different deviations of the `TraceStartTime`[4]s from the reference time must be subtracted from the delays. The resulting shifts must be applied with opposite sign: When a signal arrives later at one station then the other, its trace must be shifted back in time to match, so positive delay means negative shift.

The user can constrain the start and end of the result traces in bin numbers of the shifted traces. This was introduced primarily to save space and computation time, especially when using ascii output for debugging. As the dominant drain of computing power turned out to be the $\vec{E}$ field reconstruction, this was not honed further, i.e. to using nanoseconds. The feature is left in anyway, as it can be useful in some conditions. But the configured parameters can

[4] found in the station level event data `revt::Station`

be left usually at the default of -1, indicating that the whole input traces are used. When other values are used, the code assures that they do not exceed the available bounds.

Using the integer bin numbers as bounds also makes it clearer that the source traces start at 0, whereas the zero value in nanoseconds is ambiguous, not least because of the time offset described below. On the other hand the user has to take into account that the proportion between time and bin number changes with the upsampling factor. This seems reasonable, as these configuration parameters are seen as a more low-level intervention into the module functionality, this seems reasonable.

The reference time for the beam forming is, as described in paragraph 5.2, the core time of a preciding analysis, if present. So if this would be taken as-is as the zero time for shifting, anything happening around the core time would show "around" bin 0 in the result: Because the fourier transformation treats the signal as if it were periodically repeating, anything happening before the core time would be wrapped to near the end of the trace. So just the interesting region is split in two in this case. In the other case, without preceding analysis, the reference time is the mean trace start time, so the shower signal should already be in a sensible place, dependent on the trigger configuration.

To deal with this problem, an additional configurable time offset is applied: An additional shift of `TimeOffsetFactor` times the input trace length. So to get the core time into the middle, as optimal for the case of using the reconstructed core time, use a value of 0.5. To leave the traces as they are for the other case, set `TimeOffsetFactor` to zero. The default is 0.25, which should be ok for both situations.

The time shifting method of the module returns the offset as it appears in the *result trace*, that is, corrected by binning times startbin. This is used to "correct" the subsequent time results. So in the stored results, 0 ns reflects the reference time, and an absolute time can be retrieved from the stored core time plus the beam forming result time.

Calculating interferometry quantities When the shifted traces are stored, calculating the interferometry quantities according to the formulae is very straightforward. The only pitfall is that whilst the trace template provide binary operations on their data, applied binwise, the used functors only work with one dimensional numbers. Also there is no simple way to map a custom function to a set of traces. So both methods, for cross correlation and power, use small helper methods to iterate over one or two input traces and one output trace. This is a minor programming efford, but is not very elegant.

Both methods first allocate two trace variables of doubles, one for the sum and the other for the current addend. This is done on stack. The result, the sum divided by number of addends, is returned by-value, so also on the stack. This behaviour implies that some KB of data are copied on return, if the compiler is not clever enough to have the return value written directly into the variable that holds it. On the other hand, this scheme eliminates the possibility of memory holes. It also avoids the consequences of heap fragmentation, that can cause a bigger slowdown: Allocation on stack means usually just subtracting from the stack pointer; and copying one contiguous block of memory is very fast. Repeated allocation on heap can mean that many small portions of memory must be moved around.

If copying the result is too slow, it could be filled into a provided reference parameter, at the cost of readability, as the main routine `Run` will take care more about memory, and coding style will become more imperative.

Then both methods loop over the station traces, collecting the addends: `powertrace` iterates over all stations i, its helper method `efieldtopower` writes $|\vec{E_i}(t_j)|^2$ for every time bin j into the addend trace, which is added to the sum (see equation 6). Because the intensities are one dimensional double values, they can be summed up binwise with the trace's `+` operator, that uses the `std::plus` functor.

The `crosscorr` method loops over all pairs of stations with $i \neq j$: For each station i loop with j running from $i + 1$ to n. Its helper method `efieldproduct` takes traces i and j and writes $\vec{E_i}(t_k) \cdot \vec{E_j}(t_k)$ into the addend, according to equation 7.

The `xtrace` method works different. It uses the already existing other traces, together with their mean values that are determined by the `findPeak` method described in the next paragraph. Taking as input the cc trace, its mean, and the power trace with its mean value already subtracted. Then it applies the formula 9: It copies the cc trace into the result variable, and for each bin multiplies it with $|\mathrm{cc_i} - \langle \mathrm{cc} \rangle / \mathrm{power_i}|$.

Finding Peak The method `findPeak` uses at present a simple maximum finder to get the highest bin of the provided trace. The quantities it finds are returned in an object of the type `BeamQuantities` described in section 5.5: The result time is the bin number multiplied by the binning, with the aforementioned offset subtracted. Height is the bin value.

The other quantities are determined by means of the `utl::TraceAlgorithm` utility class: RMS is the result of the `RMS` method, the output of `Mean` is used as an estimate for the vertical offset.

5.5 Output Data

Figure 12 depicts the data structure created for the beam forming results. These were inserted into the structure that holds the reconstructed shower quantities of each event, the event variable `RRecShower` of the class `ShowerRRecData`.

The main building block is a class for the beam quantities at one sky position, named `BeamQuantities`. It holds the geometrical information of the coordinates stepped through, and three objects to hold the relevant quantities of the three traces. The geometry embraces the direction in zenith and azimuth, and the wave front curvature or cone angle. The latter are set to zero when not the associated wave model is used.

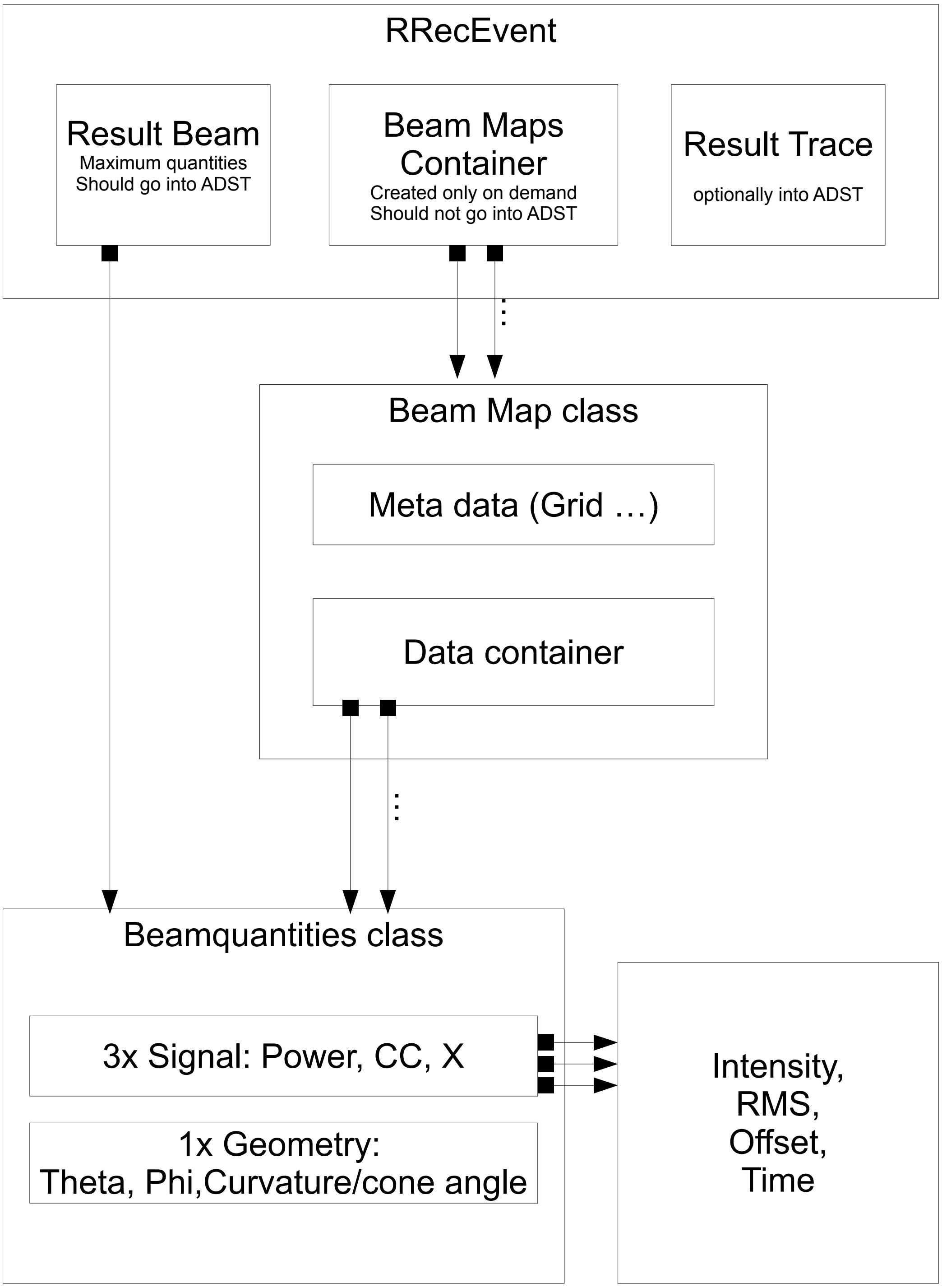


Figure 12: Schematical view of output data

The three substructure objects have the type `BeamPeak` and consist of the four double quantities sought by `findPeak`: The peak intensity and time, the trace's RMS and a possible vertical trace offset. As these objects are contained in the `BeamQuantities` object, in technical terms as object aggregation by value, the memory layout is identical as if they were stored there as individual doubles. But the code becomes more readable in many places because of the additional structuring, and the ability to hand around all quantities in one variable.

The nature of the `BeamQuantities` to hold all its data by value as a self-contained object in a continuous piece of memory becomes important when filling it into a container. It also avoids the danger of referencing memory that is overwritten in later steps.

One `BeamQuantities` variable named `BeamResultQuantities` is inserted into `ShowerRRecData` to hold the overall result. It is stored as a pointer with getter and setter methods, as well as a `HasBeamResultQuantities` to test for existence. The setter allocates new memory if the object is not existing so far, the template `utl::LameShadowPointer` assures that it is deallocated together with the `ShowerRRecData` object.

Data fields for the three beam forming traces are treated the same way: Three shadow pointer variables pointing to `utl::Trace` objects, `BeamResultCCTrace`, `BeamResultPowerTrace` and `BeamResultXTrace` with respective setters and getters were added to the `RRecShower`.

For storing beam forming information for the whole rastered sky portion, the `BeamMap` class was created. This class is a wrapper around a vector of `BeamQuantities`, exposing its iterators through this vector as `BeamQuantitiesIterator`. Further, it contains a vector of `BeamGrid` objects, which store meta information on the grid per coordinate: The minimum, maximum and step size. It may also be useful to store the coordinate name there, but this was left out for the time being.

To be prepared for a future iterative rastering, a vector of `BeamMaps` was added to the `ShowerRRecData` class, so it can store more than one map.

5.6 ADST

The main output facility and file format for reconstructed data is the *Advanced Data Summary Tree* (ADST). It uses ROOT trees to store the information in a structured way. This allows for storing complicated information in a similar representation as in memory, in a compact data format. ADST is explained in detail in [46].

ADST mirrors the whole event data structure, but the respective classes stay independent. So every class that should be stored must be written again within the ADST source tree, this time inheriting from ROOT's `TObject` class, so ROOT can handle them. After the types have made known to ADST, it parses the source files and includes the variables in the structure for the output file. These steps were done for the beam forming data described above.

To get the reconstructed data from offline's run-time representation into the ADST `RecEvent`, the module `RecDataWriterNG` was updated. It creates the RecEvent and the file to hold it, and then copies over the reconstructed data. The data trees are kept separate, and the classes usually only grant access to their quantities individually with getter methods. So the variables must be copied one-by-one in the `RecDataWriterNG` code. This is no big performance problem, but makes the code a confusing long list of calling getters and corresponding setters.

The latter was done for the result beam quantities to have the most important results in the ADST file. But it was delayed for the other data, as the adding clashed with a restructuring of the surrounding code for radio event writing. It should not take too much time to complete it though, as the additions are written as modular as possible under the preconditions.

5.7 ASCII

There is also a very primitive output mechanism producing ASCII files. It was used for initial tests before the ADST output was created. Before even types for the event structure were conceived, looking at the data was necessary. As it can be useful for debugging, or to study effects not caught by the `BeamPeak` data, it was left in.

The ASCII output writes one text file for each trace type (*cc*, *p*, *x*), with one line per value, in the format `x y z t val` respectively `r θ φ t val`, with the current direction vector in cartesian or polar coordinates, depending on the XML configuration.

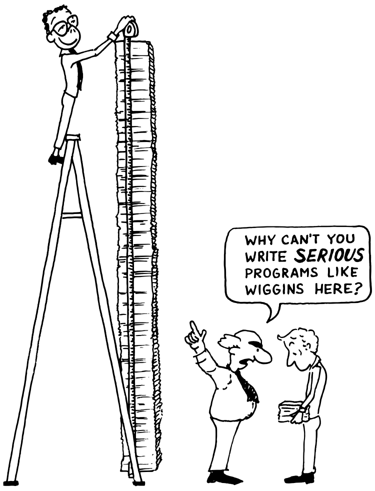


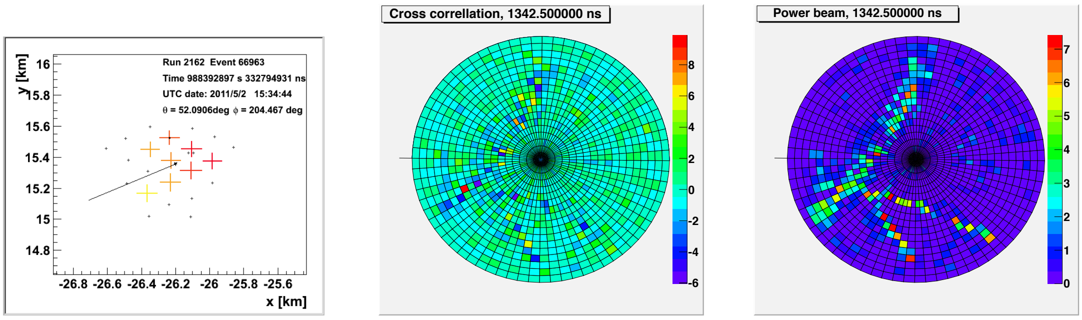
Figure 13: By relying on the abilities of the Offline framework, it was possible to keep complexity and size of physics related code at a minimum. (Sketch: [47])

6 First results

This section showcases some example runs of beamforming pipelines. The primary interest is to test and demonstrate the functionality of the method and the accuracy of the results.

The first two subsections regard directional information using a plane wavefront. First, an early 2D visualisation is shown with some of the properties of beamforming learned from it. Then coincidence events are used for validation. The third subsection examines a full 3D imaging made with the curved wavefront model.

6.1 An early run



(a) Plane fit reconstruction

(b) Cross Correlation skymap

(c) Power skymap

Figure 14: An early run for event 2162-66963

Figure 14 shows the result of an early tryout run for event 2162-66963. This is the event measured in coincidence with a surface detector event with 8 AERA stations, the highest number measured so far. For comparison: The SD arrival direction is $\vartheta = 50.7° \pm 0.5°$, $\varphi = 206.3° \pm 0.6°$, the standard radio reconstruction gives $\vartheta = 52.1°$, $\varphi = 204.8°$.

The analysis pipeline applied is the one described in section 5.1, except that there was no ADST write-out at that time. Instead, the ASCII output was redirected into a simple root script via a named UNIX pipe [48]. This script fills one 2D histogram (ϑ, φ) with the incoming intensities for every time bin. The given time, 1342.5 ns, is measured with respect to beginning of the trace. A sign flaw was found later in the trace shifting code, so traces were shifted slightly into the wrong direction. The traces are centered around the trigger information, therefore, the applied time shift is only small, resulting in a small effect of the sign error.

The run was done with a grid of 5° step size for both angles, from zenith to horizon at $\vartheta = 90°$. The intensities are shown in units of the respective trace's RMS.

The cross correlation sky map fig. 14b shows a clear single peak in the $\vartheta = 60°$, $\varphi = 205°$ bin, in crude agreement with the other reconstructions. In the power beam histogram 14c, there is a local maximum, too (in orange), but here the intensity is spread over a large area.

It has several higher maxima, which correlate to spots of high anticorrelation in the cross correlation (blue spots). The pattern in the power map are can be explained by the symmetries of the grid geometry, where different angles cause identical delays for stations on a line.

A look into the cross correlation traces shows that the maximum stays at $\varphi = 205°$ but jumps between $\vartheta = 40°$ and 60°, with a tendency to move from higher to smaller angles. This may be an effect of the sign error, but might also be overlaid with real shower movement: As the particle moves faster than c/n, but we look into signal arrival times, we should see it moving upwards. On the other hand, the difference of n to 1 is tiny for air: With $n = 1.000292$ the difference is about 1 ns per kilometer ($\Delta t = l \cdot (n-1)/c \approx l \cdot 9.7 \cdot 10^{-13}$ s/m), the maxima in the cross correlation trace are in an interval of 30 to 100 ns, depending on which peaks are taken into account, so probably the timing error predominates.

All in all the result shows that the cross correlation is suitable to locate the spot of maximal signal coherence, in coincidence with the shower direction from the other reconstruction, the power gives only a rough picture of possible directions.

The plots also show that the RMS normalization is working as expected, removing the bias from the antenna pattern: The fluctuations of the signal-less areas are uniform for all directions, and the signals behave plausibly, whilst the field estimated strength of both signal and noise grows more than exponentially towards the horizon.

6.2 Verification with SD Coincidences

To verify the beam forming machinery, an analysis according to the procedure described in section 5.1 is performed on events which coincide with SD events. These are taken from the AERA wiki[5].

The module sequence steps through the zenith and azimuth angles, does the beam forming with a plane wave front model, and selects the highest cross correlation peak (in RMS units). For this analysis, a step size of 2° ($1/90\,\pi$) was chosen for both angles, so the rounding error caused by the stepping corresponds to a standard deviation of $2°/\sqrt{12} \sim 0.6°$ ($= 1/300\,\pi$). This should be fine enough for a qualitative verification.

The reconstruction with these parameters took roughly between 70 and 100 Minutes per event on an Athlon 64 server. Table 1 summarizes the results for the five coincidence events with more than 4 radio stations, in comparison with the SD reconstruction and the standard radio reconstruction using a plane fit. As interferometry methods benefit from using more signals, the agreement should grow with the number of stations. The numbers in parentheses are the uncertainties of the SD reconstruction.

The table shows that both the beam forming result and the usual radio reconstruction deviate from the SD value by similar amounts. This can be interpreted in two ways: On

[5] at https://www.auger.unam.mx/AugerWiki/AERA%2C_Coincidence_Events

Table 1: Results for SD coincidence events

			Rd Beam		SD		Rd Std	
Rd ID	SD Event	# Stat	Zen	Azi	Zen	Azi	Zen	Azi
2162-432988	11638937	7	54	342	51.7(4)	340.5(4)	47.6	339.6
2162-66963	11629745	8	46	204	50.7(5)	206.3(6)	52.1	204.8
2162-1185999	11645902	7	20	54	63.5(3)	242.1(3)	90.0	238.4
2155-2856715	11570943	6	58	348	56.1(1)	350.4(2)	59.1	346.7
2150-1039654	11531810	5	38	296	34.7(4)	294.8(8)	37.4	294.4

the one hand one can call the method verified, as its result are at least not worse than the standard reconstruction and show a similar behaviour. On the other hand, the results lie clearly outside of the error interval and show no clear advancement towards the standard reconstruction that costs only a small fraction in computing power.

For event 2162-1185999 the beam result points in a totally different direction. But comparison with the standard reconstruction shows: (a) the plane fit result also has a big deviation towards SD, (b) it even is at 90° at the horizon, whilst the beam analysis goes only up to 80°. So there are several options: The coincidence could be false. The maximum could really be at 90°, but the analysis cannot find it because of the limit. The whole event could be something man-made or a superposition of noise and ”real” event. In any case, the big deviation of the standard reconstruction justifies not taking this event too serious.

To get a measure for the deviation from SD, we regard the angular residual

$$\Omega = \arccos(\vec{e}_1 \cdot \vec{e}_2) \qquad \vec{e}_i = (\sin\vartheta_i \cdot \cos\varphi_i, \sin\vartheta_i \cdot \sin\varphi_i, \cos\vartheta_i) \tag{15}$$

that gives the angle between the directions, here expressed as unit vectors $\vec{e}_i$.

Table 2: Angular residuals towards SD reconstruction

Rd ID	SD Event	#	Ω_{Beam}	Ω_{Std}
2162-432988	11638937	7	2.6	4.2
2162-66963	11629745	8	5.0	1.8
2162-1185999	11645902	7	83.3	26.8
2155-2856715	11570943	6	2.8	4.3
2150-1039654	11531810	5	3.4	2.7

Table 2 confirms the impression from table 1: Deviation from SD are similar for both radio reconstructions. Unfortunately the benefit of higher station number cannot be seen: Event 2162-66963 with 8 stations even shows the highest deviation.

The inaccuracies of the station clocks are seen as the dominant error sources. From a geometrical point of view, an uncertainty of 20 ns means a deviation of $\arcsin(20\mathrm{ns}\cdot c) \approx 2.3°$; for the worst case of a vertical shower hitting two stations with 150 m distance. But the cross correlation is highly non-linear. So the real effect of timing errors to the reconstructed angle cannot be derived that easy. Probably the best way to get hold of them would be numerical simulations.

6.3 3D Reconstruction

To try out full 3D reconstruction, again event 2162-66963 is used. With the most stations it should contain the most curvature information and yield the best results. The sequence was run with a cartesian raster centered over the unweighted station mean. The parameters are

Table 3: Parameters of example 3D reconstruction

stepsNorthSouth:	100	distanceNorthSouth:	20.0 km	startBin:	512
stepsEastWest:	100	distanceEastWest:	20.0 km	stopBin:	1536
stepsHeight:	50	minHeight:	3.0 km	upSampling:	none
		maxHeight:	13.0 km		

summarized in table 3. Note that 100 steps here means 101 data points in a row, so that one step is 0.2 km and the grid is still going symmetrically and inclusively from corner to corner.

The trace length limitation is to lower the data rate by a factor two, the maximum will surely be in this region. Also, no upsampling is done, as here finding the absolute maximum has no priority. Nonetheless, calculation took 6.9 days of pure computing time on the Athlon 64 workgroup server – about two weeks of real time, as processor share fluctuated between 30 and 80%. 55% of the time was consumed by the $\vec{E}$ field reconstruction, 45% by the beamforming module.

Apart from the relatively high resolution of the grid and the scanning of regions far from the expectation, that can be left out when the method is trusted enough, there are other “wastes” of computation time which can be avoided in production: Because of the cartesian grid, the arrival angles are different for every point, so the antenna sensitivity has to be looked up and interpolated again. In a “real” run, a polar grid should be applied, that holds each direction constant while scanning through the distances, so antenna response can be cached. The transformation to cartesian coordinates can be deferred until visualisation.

These shortcomings are accepted deliberately not to anticipate expected results but to have the machinery do the work by itself.

Figure 15 shows a 3D plot of the resulting cross correlation for random noise at the first time bin, far away from the signal. It demonstrates: The curvature sensitivity is so small, that all signals, even noise fluctuations, are stretched to long lobes, which the willing eye

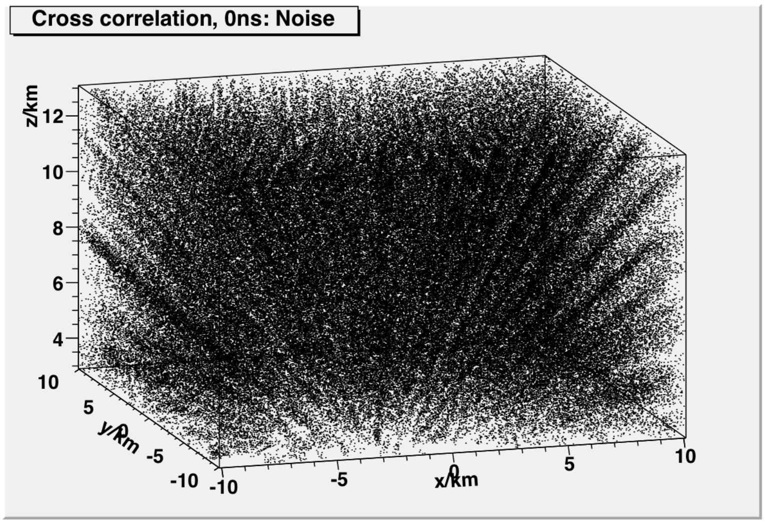


Figure 15: Noise far from the event time (3D Reconstruction of event 2162-66369)

may take for a trajectory towards the center. This must be kept in mind when interpreting a beam forming picture. With even more stations, the sensitivity should be better, but only gradually, as the wave front footprint, and thus the baseline of stations recording it, is much smaller than the radius. As the grid is cartesian, this cannot be a grid or coordinate transformation related artifact.

The figure also shows that in a scatter plot, any noise will make the signal hard to see, by the sheer number of scatter points around it. So it must be suppressed to carve out the signal.

The time bin with the highest cross correlation normalized to RMS of 21,4 is shown in figure 16. The histogram was only filled with values above 17. The regions outside of the zoom are blank with this condition; the noise and side lobe intensity there is in the order of 5.

As said, one cannot state in how far the long drawn-out shape is caused by the distance ambiguity, or by real source extent. The two dimensional projections suggest a rough compliance with the reconstruction angles of the event: A quick slope estimation gives a quite exact congruence for the incoming azimuth (from the top view) of

$$\varphi \approx \arctan(4.5/10) + 180° = 204.2° \tag{16}$$

but it seems a little bit more inclined from the zenith (estimated from the side views)

$$\vartheta \approx 90° - \arctan\left(8/\sqrt{10^2 + 4^2}\right) = 53.4° \tag{17}$$

Although the real shape of the source cannot be derived from the images, there is a clear radius dependency of the signal. So at least the maximum radius should be trustworthy. The

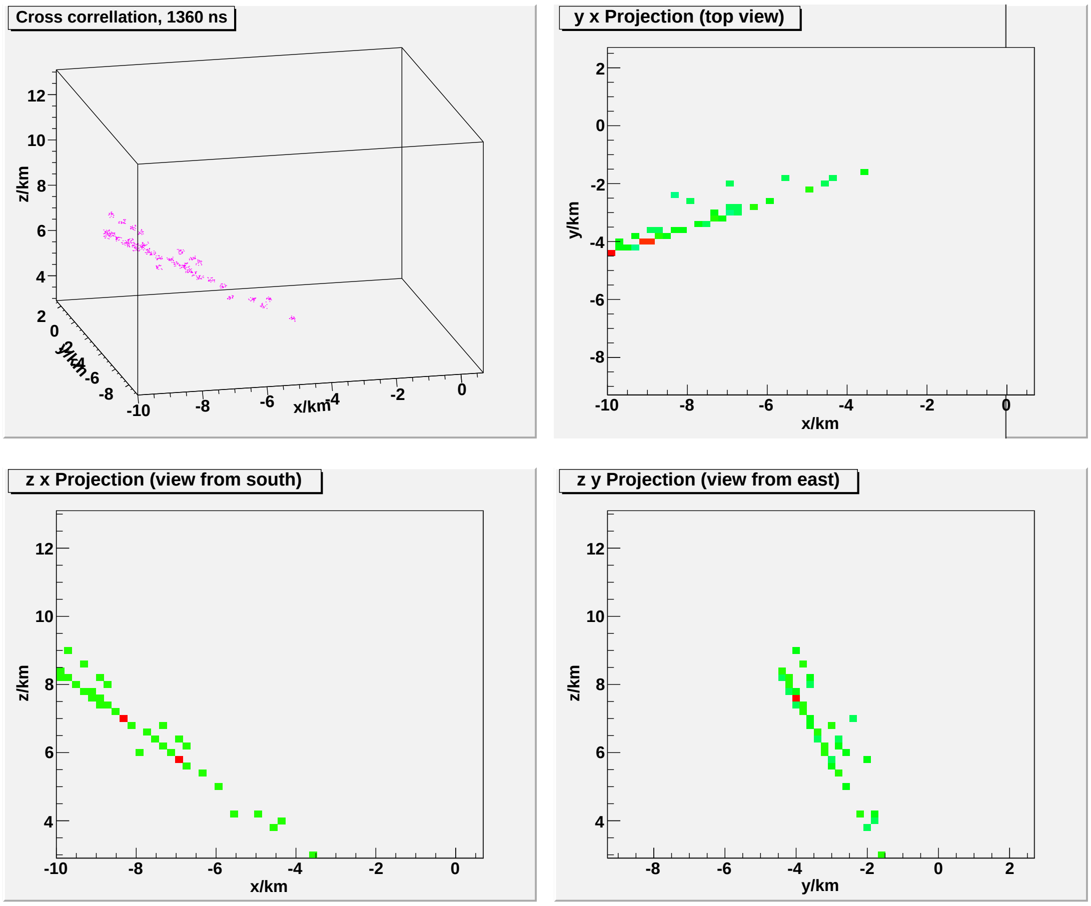


Figure 16: 3D Reconstruction of event 2162-66369

maximum bin is at the position

$$x_{\mathrm{max}} = -8.8\,\mathrm{km} \qquad y_{\mathrm{max}} = -3.8\,\mathrm{km} \qquad z_{\mathrm{max}} = 7.4\,\mathrm{km} \tag{18}$$

in spherical coordinates

$$\varphi_{\mathrm{max}} = 203.4° \qquad \vartheta_{\mathrm{max}} = 52.3° \qquad r_{\mathrm{max}} = 12.1\,\mathrm{km} \tag{19}$$

Comparing this with table 1, the angles are quite near the standard radio reconstruction, a little off from the SD values. Why the plane wave beam forming deviates much more there, is not clear. It may have to do with the side lobes that are visible next to the line of highest intensity. Of course for studying the maximum coordinates in practice, a polar grid is the only sensible tool of choice.

It would be interesting to repeat analyses like these on events with more stations. So far verification was the focus, which requires SD coincidences. In addition, it should be beneficial to include also untriggered stations for beamforming. Trying beam forming on signals from a known source, i.e. with the octocopter, would be an idea to get a better feeling for the radius sensitivity. On could already investigate it by excluding stations iteratively from the analysis; but there should be a higher number to start with.

Meanwhile, there are simulated events waiting to be analyzed. Their advantage is the knowledge of the exact input parameters. But the emission mechanisms are known only with uncertainties, so I regarded them second for initial validation.

7 Alternative Idea

This section presents an alternative way to use beam forming methods to analyze AERA data. This idea came up quite late in my studies and was only examined superficially. On the other hand, the implementation is fully functional and needs only verification, and maybe some parameter tuning.

7.1 Approach

The idea is to reverse the flow of information used in the “usual” approach. There we calculated signal arrival delays from a physical model, scanning through all possible source directions in the sky. Signals were shifted according to these delays before calculating the interferometry quantities. Then a maximum voxel was selected.

Here we try to maximize the output, the cross correlation peak, by varying the time shifts. From the ”optimal” shifts incoming delays are calculated and as reconstructed arrival times made available for further analysis. In effect the module can be seen as a replacement for `RdStationSignalReconstructor` in the module sequence.

A benefit of this approach is that the usual modules for direction reconstruction for different wave models can be used—now with input data that includes the phase information instead of merely the maximum bin. Another point would be visualizing the delays to explore the shape of the wave, instead of postulating a certain behaviour. Again the coherence timing might be more significant than the arrival times of the maximum.

As a downside, this method can not produce pictures of a sky region, but reduces signal information to one number per station. It could not be verified if the additional computation leads to a significant improvement of later analysis.

7.2 Implementation

The module for this method, working title `RdBeamTimeOptimizer`, was forked from `Rd-BeamFormer`. Probably the modules should be unified again, if both methods turn out to be useful. But for the experimental stage it seamed safer to keep it separate.

This section describes its steps when `Run`, in chronological order. An example analysis pipe line follows. This module uses Minuit2 with the API described in [49].

Setting initial shift values If the event already contains an `RRecShower` structure as a result of a preciding analysis, its `Axis` vector is used to obtain initial values for the shifts. Acting like the original `RdBeamFormer`, we use the `WaveModel` (configured in the `RdBeamTimeOptimizer.xml` file) to get the delays per station position. These are applied to the difference of `TraceStartTime` and reconstructed core time, and thus converted to shifts. In contrast to `RdBeamForming` this is only done once in the beginning, so the analysis is not fixed to this wave model.

An additional shift of half a trace length is added to get the expected peak roughly in the middle of the result trace. Otherwise the core time would correspond to the first bin. Anything distributed around it would jump between the beginning and the end of the trace, because of the periodicity of the FFT-based shifting.

If no preceding analysis is found, all shifts are initially set to zero. For self-triggered station data, this corresponds to a time stamp fit with the "true" wave model. For externally triggered stations to a plane wavefront fit of a signal from the zenith. One might argue that it would be more consistent to begin with the delays of wave coming from zenith, or the same angle that `RdAntennaChannelToStationConverter` uses initially. But it seems compelling to start at "truth" and being able to start without a physical model at all.

With these presets, the module is capable of both fine-tuning an existing analysis and serving as timing reconstructor in a iterative analysis, in place of or addition to `RdStationSignalReconstructor`.

Minimizing parameters With these start parameters, the optimal shifts Δt_i are determined in two steps: First, the normalized power $\frac{p(\vec{\Delta t})}{\mathrm{RMS}_p(\vec{\Delta t})}$ is maximized. Then with the resulting set $\vec{\Delta t}$ of parameters as new start values, the same is done for the cross correlation $\frac{cc(\vec{\Delta t})}{\mathrm{RMS}_{cc}(\vec{\Delta t})}$.

Minuit expects a function to be minimized, the FCN, implemented by a class inheriting from `FCNBase`, with the operator `()` taking a vector of input variables `std::vector<double>` and returns the respective function value. Here, two such classes are created which only differ in calling `powertrace` respectively `crosscorr`. To call these methods from the module class, the FCN objects hold pointers to the `RdBeamTimeOptimizer` object, set in the constructor.

So the FCN operator code is very simple: Call back to the module to create the respective trace with the given shifts. Then call `findPeak` and return its intensity divided by RMS, negated because minuit minimizes the FCN. The `findPeak` method was changed to cache its results in the module, so that they do not have to be calculated again with the optimal shifts. This circumvents the constant nature of the FCN class required by minuit, not very elegant but effective.

To do the minimization, a `MnMinimize` object is created from the respective FCN object and than run, returning an object representation the minimization result. From this, only the new parameters are copied to the parameter variable.

Finally, the optimized time shifts Δt_i are converted to per-station arrival times t_i by

$$t_i = \Delta t_i + t_{\mathrm{peak}} - t_{\mathrm{offset}} + \texttt{traceStartTime}_i \tag{20}$$

and written into the station level reconstruction data signal time.

7.3 Results

As an initial test, the SD coincidence events are used again. The module sequence consists of an initial iterative plane fit loop, identical to the standard reconstruction. Then the RdBeamTimeOptimizer runs to improve the timing information, with the reconstructed $\vec{E}$ field from the last iteration. Finally, the plane fit is applied again with the updated timing.

A run for one event took 3 to 5 minutes, with RdAntennaChannelToStationConverter and RdBeamTimeOptimizer using a similar amount of computation time of about 48 %. Table 4 shows the resulting angles, compared with SD and the standard reconstruction. It seems that the beam optimizing does not greatly change them. This may mean, that the former result was already very near the optimum with phase information taken into account. But it may also be the case that the optimization falls into a local minimum near the start parameters.

Table 4: Beam optimization results for SD coincidence events

			SD		Rd Std		Rd BeamOpt	
Rd ID	SD Event	# Stat	Zen	Azi	Zen	Azi	Zen	Azi
2162-432988	11638937	7	51.7(4)	340.5(4)	47.6	339.6	47.5	339.2
2162-66963	11629745	8	50.7(5)	206.3(6)	52.1	204.8	51.5	204.4
2162-1185999	11645902	7	63.5(3)	242.1(3)	65.8	241.6	65.5	241.6
2155-2856715	11570943	6	56.1(1)	350.4(2)	59.1	346.7	57.2	347.7
2150-1039654	11531810	5	34.7(4)	294.8(8)	37.4	294.4	36.7	294.6

Table 5: Angular residuals towards SD reconstruction

Rd ID	SD Event	#	Ω_{Std}	Ω_{BeamOpt}
2162-432988	11638937	7	4.2	4.3
2162-66963	11629745	8	1.8	1.7
2162-1185999	11645902	7	2.3	2.1
2155-2856715	11570943	6	4.3	2.5
2150-1039654	11531810	5	2.7	2.0

Again we regard the angular residuals in table 5. On average they disclose a small advance towards the SD reference. This is a promising result and should be pursued further, as it might be easy to improve with better optimization settings.

Interestingly, the biggest improvement is seen for the events with least stations, although a handful of events is not sufficient to draw conclusions. It may be a coincidence that station timing varies least between the involved stations, or that the events are recorded at better weather conditions. On the other hand, the curvature plays a smaller role for less stations, so a plane fit may profit from the time optimization less at events with more stations. The method should be tried out with the upcoming modules for curved or conical wave fronts for the final fit.

Note that the deviations are roughly compatible with the above estimate of 2.1° from 20 ns timing error in geometrical reconstruction.

7.4 Problems

It is conspicuous that the results after the beam optimization are very close to the usual plane fit, situated between SD and Rd reconstruction. This has some physical sense, and one could argue that there is a certain deviation between SD and radio signal direction, especially when both radio reconstructions use the same fit module with plane wave approximation, and suffer from identical timing offsets from the GPS clocks.

On the other hand, the small change could be caused by the optimization process' catching a local maximum in the near of the start parameters. Inspecting the minimization execution in GDB shows that times are mostly only changed by a few nanoseconds at most, in spite of an initial step size of 100 ns (even a few 1000 ns were tried out). From a physical point of view, also this is sensible; differences between wave crest based timing and best phase coherence should be smaller than the periods.

But to test the robustness of the results, the shifts were manipulated by hand in random directions. It turned out, that it was quite easy to get minuit to find unphysical values by adding offsets of a few hundred or thousand ns. This is only a technical weakness, as the minimized FCN value grew dramatically (to less negative values). If this is a concern, maybe the errors should be scaled bigger again after the power maximization; or several minuit runs should be done with varying start parameters.

For smaller changes in the region of the 20 ns timing errors, the resulting optimal shifts were observed to be robust, varying under 0.1 ns. This speaks in favor of the correctness, but one should have an eye on the affair.

Different FCNs were also tried out, i.e. by taking the mean of the trace around the peak to smoothen the function, but these attempts did not improve the robustness.

8 Conclusion

Two methods of beam forming were implemented into Offline and subjected to initial test runs, with promising results. Both methods are in a usable state, but can be honed further to broaden their scope of application.

Stepping through wave parameters The “usual” beamforming method scans through the parameters of the wave model, like the source position. It is applied at the Lopes experiment, but was completely written from scratch for Offline, using full 3D $\vec{E}$ field quantities.

This method was successfully applied to create 2D and 3D visualisations of the radio intensity distribution in the atmosphere, and for directional reconstruction.

For validation, events measured in coincidence with the surface detectors were used. Both direction reconstruction and imaging show a good compatibility with the SD events. The existing small deviations can be explained by inaccuracies of the AERA station timing.

The 3D visualizations contain not only information about the direction but also the distance of the source. The exact shape can probably not be seen, because the distance resolution is smaller than the angular one. But the images show a clear intensity variation with the radius. Also this resolution will improve when the timing is corrected.

It would improve all beamforming measurements to have more signals, i.e. from untriggered stations. Especially the distance sensitivity based on the wavefront curvature, should benefit.

In the beamforming pipeline, relatively simple additions to the grid module will allow saving computation time by selecting a region of sky in a previously reconstructed direction, or by applying recursively finer grids.

Improving fit reconstructions with phase sensitive signal times This method determines the relative timing of the signals that leads to the highest coherent intensity. This is used to get a better estimate on the signal flight duration than the peak alone. With the usual fit models the final (direction) reconstruction is performed.

To my knowledge this method is completely new, at least in the context of radio air shower measurements. So far only a few initial reconstructions were run; with encouraging results: For SD coincidences, the output moves slightly but notably towards the SD angles.

This method is many times faster than the raster method, with the current unoptimized grid modules by orders of magnitudes, but does not offer its broad spectrum of capabilities. The functionality of the implementation is complete, further studies should be made on its robustness of finding the main maximum.

In summary, two improved methods of reconstruction for radio data containing electric field information were developed for the offline software framework, with different advantages. They can be used to analyse and visualise radio shower events measured by AERA, for studying radio emission mechanisms and shower evolution. This, especially with events measured in coincidence with the colocated surface and fluorescence detector, will hopefully lead to a deeper understanding of extensive air showers and cosmic rays.

“I still don’t see how these programming conventions enhance readability.” [47]

References

[1] V. Hess *Physikalische Zeitschrift*, 13:1084, 1912.

[2] Alessandro De Angelis *L'enigma dei raggi cosmici* Springer 2011.

[3] Claus Grupen *Astroteilchenphysik: Das Universum im Licht der kosmischen Strahlung*, Vieweg 2000.

[4] J. P. Wefel *Cosmic Rays, Supernovae and the interstellar Medium* volume 337 of NATO ASIC Proc., page 29, 1991

[5] K. Nakamura et al. (Particle Data Group), J. Phys. G 37, 075021 (2010)

[6] K. Greisen. *Phys. Rev. Lett.*, page 748, 1966.

[7] G. T. Zatsepin and V. A. Kuzmin *JETP Lett.*, 4, 1966

[8] N. Hayashida et al. *Updated AGASA event list above* $4 \cdot 10^{19}$ *eV* Astron.J. 120 (2000) 2190 [arXiv:astro-ph/0008102]

[9] The Pierre Auger Collaboration *Measurement of the energy spectrum of cosmic rays above* 10^{18} *eV using the Pierre Auger Observatory*, Phys.Lett.B685:239-246 2010 [arXiv:1002.1975v1]

[10] R.U. Abbasi et al. *Measurement of the Flux of Ultra High Energy Cosmic Rays by the Stereo Technique* Astropart.Phys.32: 53–60, 2009 arXiv:0904.4500v1

[11] E. Fermi *Phys. Rev.*, 75:1169, 1949.

[12] Gaisser, T. K.: Cosmic rays and particle physics. Cambridge University Press, 1990.

[13] Pierre Auger *Extensive cosmic-ray showers* Rev. Mod. Phys. 11, 288-291, 1939.

[14] O. C. Alkhofer *Introduction to Cosmic Radiation* Verlag Carl Thiemig, 1975.

[15] Gurgen. A. Askaryan. Soviet Phys. JETP, 14:441, 1962.

[16] J. V. Jelley, J. H. Fruin, N. A. Porter, T. C. Weekes, F. G. Smith, and R. A. Porter. *Nature*, 205:327, 1965.

[17] H. R. Allan. *Prog. in Element. part. and Cos. Ray Phys*, Vol. 10:171, 1971.

[18] F. D. Kahn and I. Lerche. *Radiation from Cosmic Ray Air Showers* Royal Society of London Proceedings Series A, 289:206–213, January 1966.

[19] H. Falcke and P. W. Gorham *Detecting Radio Emission from Cosmic Ray Air Showers and Neutrinos with a Digital Radio Telescope* Astropart. Physics, 19:477, 2003.

[20] C. W. James, H. Falcke, T. Huege and M. Ludwig *General description of electromagnetic radiation processes based on instantaneous charge acceleration in 'endpoints'* [arXiv:1007.4146]

[21] Andreas Haungs for Marianne Ludwig and Tim Huege *Latest development of air shower radio emission simulated with REAS* Auger Collaboration Meeting, November 2011 [slides]

[22] Olaf Scholten, Krijn de Vries, Klaus Werner *High-frequency components in radio signals; Cherenkov effects* Auger Collaboration Meeting, November 2011 [slides]

[23] Exploring the Auger Observatory with Google Earth `http://www.phys.psu.edu/%7Ecoutu/Auger_Google_Earth.htm`

[24] A. Etchegoyen et al. for the Pierre Auger Collaboration *The Pierre Auger Project and Enhancements* [arXiv:1004.2537v1]

[25] auger.org image gallery

[26] Luis A. Anchordoqui, for the AUGER Collaboration *The Pierre Auger Observatory: Science Prospects and Performance at First Light* arXiv:astro-ph/0409470v1

[27] A. Haungs *Air Shower Measurements with the LOPES Radio Antenna Array* Nucl. Instrum. Meth. Suppl. A 604, 1-8, 2009.

[28] P. Lautrido *Development of a radio-detection method array for the observa- tion of ultra-high energy neutrino induced showers* Nucl. Instrum. Meth. Suppl. A 604, 85-87, 2009.

[29] The AERA group *AERA proposal for the construction of the 20 km^2 Auger Engineering Radio Array* GAP-NOTE 2009-172

[30] Jens Neuser *Radio Measurement of Extensive Air Showers at the Pierre Auger Observatory*

[31] M. Erdmann *Antenna alignment for the frst 24 stations of AERA* GAP 2010-083, 2010.

[32] M. Stephan *Design and test of a low noise amplifier for radio antennas mea- suring atmospheric showers* GAP 2010-045, 2010.

[33] A. Horneffer, Dissertation University of Bonn, *Measuring Radio Emission from Cosmic Ray Air Showers with a Digital Radio Telescope*, 2006.

[34] Joseph. W. Goodman, *Statistical Optics* 1985 [ISBN: 0471015024]

[35] A. Nigl et al. - LOPES Collaboration *Direction identification in radio images of cosmic-ray air showers detected with LOPES and KASCADE* Astronomy & Astrophysics, 487:781–788, 2008.

[36] A. Horneffer et al. - LOPES Collaboration *Primary Particle Energy Calibration of the EAS Radio Pulse Height* In Proceedings of the 30th ICRC, Merida, Mexico, volume 4, pages 83–86, 2007.

[37] F. Schröder *Instruments and Methods for the Radio Detection of High Energy Cosmic Rays,* 2010. [urn:nbn:de:swb:90-223139]

[38] Michael Konzack et al. *Specifications of the AERA Beacon and its Software* [GAP 2010-099]

[39] F. G. Schröder et al. *New method for the time calibration of an interferometric radio antenna array* [arXiv:1002.3775v2]

[40] Frank Schröder, *Personal Communication on AERA Station Timing*, 2011

[41] S. Argiro, S.L.C. Barroso, J. Gonzalez, L. Nellen, T. Paul, T.A. Porter, L. Prado Jr., M. Roth, R. Ulrich, D. Veberic *The Offline Software Framework of the Pierre Auger Observatory* [arXiv:0707.1652]

[42] Pierre Auger Collaboration *Advanced functionality for radio analysis in the Offline software framework of the Pierre Auger Observatory*, [arXiv:1101.4473v2]

[43] L. Bähren, H. Falcke, A. Horneffer, S. Lafebre, and A. Nigl *The LOPES-tools Software Package Manual*

[44] L. Bähren et al. *Source code of Lopes Tools* available from `http://lus.lofar.org/wiki/doku.php?id=software:packages:lopes-tools`

[45] Oliver Krömer, Hartmut Gemmeke *Simulation of antennas proposed for radio detection of UHECRs* GAP Note 2009-035

[46] Ioana C. Mariş, Fabian Schüssler, Ralf Ulrich and Michael Unger *Data Summary Trees and Shower Visualization for Reconstructed Auger Events* [GAP Note 2006-081]

[47] Leo Brodie *Thinking Forth* ISBN 0-9764587-0-5, `http://thinking-forth.sourceforge.net/`

[48] Andy Vaught *Introduction to Named Pipes* Linux Journal, Sep 01, 1997

[49] Fred James, Matthias Winkler *Minuit User's Guide* `http://seal.web.cern.ch/seal/documents/minuit/mnusersguide.pdf`

Acknowledgement

The support of many people smoothed the way to this thesis and made it even possible. To them I owe gratitude.

First I want to mention Prof. Karl-Heinz Kampert, who gave me the opportunity to work on this interesting topic, for his friendly mentoring. And Prof. Klaus Helbing for being the second reviser.

Big thangs also go to Julian Rautenberg, who always had open ears, answers and suggestions. In fact, most ideas took shape in long discussions with him.

Thanks to my colleagues in the astroparticle physics group. When needing help, everybody was willing to spend their time for me. Especially Jens Neuser and Daniel Kruppke-Hansen explained and assisted a lot.

From other institutes, particularly Tim Hüge, Frank Schröder, Benjamin Fuchs and Maximilen Melissas supplied their knowledge and provided answers to my questions.

I also will not forget, that my studies would have been impossible without the support from my family. My wife, my parents-in-law and my friends spend a lot of time with the children to unburden me.

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Wuppertal, den _______________ _________________________