MAGIC - Roadmap to a standard analysis

Thomas Bretz
for the MAGIC collaboration

Institut für Theoretische Physik und Astrophysik, Universität Würzburg, Am Hubland, 97074 Würzburg, Germany

Abstract. The current generation of Imaging Air Cerenkov Telescopes takes data at a rate of up to 1 kHz requiring a maximum storage rate of 30 MB/s or 1.8 GB/h. Processing these data necessarily involves a lot of automation and optimisation of the software. This enforces the development of a stand-alone standard analysis. The final goal is having highly compressed output which can be provided as the full-physics input to a more dedicated analysis by collaborators or guest observers.

OVERVIEW

The current generation of large (mirror-) area Imaging Air Cerenkov Telescopes (IACT) have been build to discover new gamma-ray sources and to do precision studies of known sources. Since most sources are expected to be extragalactic and at high redshifts, as in other astronomical domains, attenuation by pair production is expected to cut off their spectra around 30 GeV to 100 GeV. The number of detectable sources should thus increase with lowering the energy threshold of the telescopes below 100 GeV The detection of lower energy gamma rays requires larger telescopes and higher quantum efficiencies of the PMT camera, increasing the sensitivity also for the steeply rising background of hadrons, muons, electrons, night sky, and faint stars. The trigger rate therefore goes up tremendously by lowering the threshold, and new techniques must be developed to detect the gamma rays. Information loss during preprocessing must be kept under control, since the analysis methods are not yet fully matured. The MAGIC telescope was designed to handle a maximum trigger rate of 1 kHz, which corresponds to a data-storage-rate of 1.8 GB/h.

Processing this amount of data needs a lot of automation and optimisation of analysis procedures done manually in past experiments. (For example: Cut optimization for different night-sky-background conditions) The importance of this step is the combination of automation and flexibility. The development of a standard analysis to handle this data should result in programs which are fast enough for automatic data-processing but still flexible enough allowing students the development of new analysis methods inside an existing framework. The advantage is the build-in compatibility of intermediate data which has been produced with new analysis methods with all other existing programs.

More information at http://magic.astro.uni-wuerzburg.de/mars
To ensure both the Magic analysis and reconstruction software (MARS) is build out of different layers (Fig.1):

**Kernal**

The Kernal\(^1\) of MARS is mainly a framework for event based analysis. The Kernal itself is C++ based (containing a lot of base-classes) and is programed on top of CERN’s ROOT-package (http://root.cern.ch). The package contains tools for general data-access, which allow the setup of data-access (such as standard cuts) from a resource file without previous implementation in the source code. This easy data-access can also be used as an input to black-box methods (Random Forest, Neural Nets, etc.). With this method, testing the signal-/background-seperation power of new variables describing the data can easily be done.

Another important part of MARS is the I/O system which allows reading and writing

---

\(^1\) The spelling of Kernal is taken from the manual of the COMMODORE 64 for nostalgic reasons
data-containers to and from a file without the need of a new code implementation. It also allows to synchronize several data branches containing data from different detector subsystems, which have been written asynchronously.

All this is put together in a general structure used to setup and run eventloops: reading, processing and finalizing data event by event (Fig.3). The generalized eventloop has a lot of maintainance features, a graphical interface (which is easy to use and program), an automatic call-counter for each task, which is processing events, run-time analysis and a root-based SQL database interface.

By having an interface to resource files, the setup of all tasks put together in an eventloop is easy.

In addition a streaming and logging system was created which allows (using C++ streaming classes) to redirect the console-logging-output at the same time to several targets (standard out, file, html). By setting debug-levels the user can easily decide, what the level of information shall be provided on the screen (Fig.3). This allows a general log-commandline-interface for all MARS based programs, the coloration of the output using ANSI color codes and the creation of plugins for all kind of output (html, TeX, ...).

The Kernel is completely application independent. It is possible to use it for new applications beyond IACTs.

**The second layer**

In a second layer general analysis methods (Neural Nets, Random Forest, ...), astrometric algorithms, an xephem-like star-display, and a lot of general physics specifics and mathematical algorithms (significance calculation, ...) are combined. An xephem-like star-display can be seen in figure 2, where a 2D sky-map (large black dots) is overlayed with a camera display.

This layer is still totally independent on the application, i.e. layers one and two can be used with almost any event-based experiment.

**IACT Standard Algorithms**

In the third layer algorithms are combined which are commonly used in the analysis of IACT data, e.g. image cleaning, image parameter calculation (*Hillas-Parameters*), flux calculation and many other tools. As this layer is still experiment independent, it is easy to adapt the existing code to a different telescope geometries (e.g. MAGIC-II, H.E.S.S., VERITAS,...).

An example of the output of such an algorithm can be found in figure 2.
FIGURE 2. A display of the three brightest stars near the Crab nebula overlayed on top of the MAGIC camera display. The black big black dots are the star positions for an ideal telescope (calculated by a 2nd layer class, overlayed on top of the MAGIC camera (taken from layer 4), displayed together with small dots which are reflections of each of the individual MAGIC mirrors (Layer 3)

Detector Geometry

The fourth layer assigns a telescope geometry (camera geometry) to underlaying layers. Adapting the existing codes and algorithms for analysing codes of different telescopes would require a new implementation of this layer.

An example of a detector geometry can be found in the star-display in figure 2.

Top-layer

In the top layer of MARS, typical experiment-specific algorithms are found, in this case for the MAGIC telescope. Examples are the PMT calibration or an online display which is run at the telescope site using the standard analysis algorithms.
CONCLUSION

The standard analysis software for MAGIC is based on the MARS package. Distinguishing strictly between the various single layers in the programming structure allows to apply the software with complete flexibility. The event-based kernal of MARS could be used generally in event-based applications such as the analysis of accelerator or space experiments. The IACT layer opens the possibility to crosscheck data analysis results obtained with different IACT systems (e.g. H.E.S.S., VERITAS, CANGAROO, MAGIC). MARS could also be used for non-event-based applications. In particular, the user-friendly display with several tabs, the logging system, and a large number of I/O features, as it is already in use for the MAGIC tracking control, may find a wide range of applications.

ACKNOWLEDGMENTS

I want to acknowledge the german BMBF (05 CMOMG1/3).
FIGURE 3. Schematic overview of the kernel structure of MARS. The left part shows a scheme of the event driven Kernel. The left part shows the graphical user interface (GUI) displaying information about a running loop, the lower part shows the (colored) console logging and the logging redirected to the GUI.