Luiza: Analysis Framework for GLORIA

Aleksander Filip Żarnecki[1], Lech Wiktor Piotrowski[1,2], Lech Mankiewicz[3], Sebastian Małek[4]

[1] *Faculty of Physics, University of Warsaw, Hoża 69, 00-681 Warsaw, Poland*

[2] *Presently at RIKEN Advanced Science Institute, Wako, Japan*

[3] *Center for Theoretical Physics, Al. Lotnikow 32/46, 02-668 Warsaw, Poland*

[4] *National Centre for Nuclear Research, Hoża 69, 00-681 Warsaw, Poland*

Corresponding author: zarnecki@fuw.edu.pl

Abstract

The Luiza analysis framework for GLORIA is based on the Marlin package, which was originally developed for data analysis in the new High Energy Physics (HEP) project, International Linear Collider (ILC). The HEP experiments have to deal with enormous amounts of data and distributed data analysis is therefore essential. The Marlin framework concept seems to be well suited for the needs of GLORIA. The idea (and large parts of the code) taken from Marlin is that every computing task is implemented as a processor (module) that analyzes the data stored in an internal data structure, and the additional output is also added to that collection. The advantage of this modular approach is that it keeps things as simple as possible. Each step of the full analysis chain, e.g. from raw images to light curves, can be processed step-by-step, and the output of each step is still self consistent and can be fed in to the next step without any manipulation.

Keywords: Telescope network, image processing, data analysis.

1 Introduction

GLORIA[1] (GLObal Robotic-telescope Intelligent Array) is an innovative citizen-science network of robotic observatories, which will give free access to professional telescopes for a virtual community via the Internet. The GLORIA project will develop free standards and tools for doing research in astronomy, both by making observations with robotic telescopes and by analyzing data that other users have acquired with GLORIA and/or from other free access databases, e.g. the European Virtual Observatory. Dedicated tools will be implemented for designing and running so called off-line experiments, based on analysis of available data. Many different types of experiments are considered, for example classification of variable stars, searches for optical transients, and searches for occultations of stars by solar system objects.

One of the challenges we have to face in designing the environment for off-line GLORIA experiments is how to deal with huge amounts of data and a large variety of analysis tasks. We need an analysis framework that will be both very efficient and very flexible. These requirements are new to astronomy. However, High Energy Physics experiments have long years of experience of dealing with enormous amounts of data and complicated analysis tasks. Experiments at the CERN Large Hadron Collider read information from about 100 million electronic channels, which is equivalent to taking 100 MPixel image of the detector, every 50 ns (20 million times per second). Even after very strong (10^{-5}) on-line selection of events (using multi-level trigger systems) GBs of data are stored every second. Data analysis for LHC experiments has to be performed on the LHC Computing Grid (WLCG), which currently includes about 170 000 TB of disk space and CPU power of about 1 800 000 HEP-SPEC06 units. However, this analysis is only possible thanks to custom-designed, highly efficient analysis software.

Detectors at the future International Linear Collider (ILC), which is the next generation e^+e^- collider under study, will deal with even larger "images". Although the project will not be realized before 2020, detailed studies of physics and detector concepts, and also detector prototype tests, have been under way for many years. Large samples of Monte Carlo data have already been generated to test detector performance and analysis methods. A dedicated Marlin[2] framework has been developed for efficient data reconstruction (corresponding to image reduction in astronomy) and analysis. We decided to adopt this framework for the needs of data analysis in GLORIA.

2 Basic concept

Marlin (Modular Analysis and Reconstruction for the LINear collider) is a simple modular application framework for developing reconstruction and analysis code for ILC. Data reconstruction and analysis should be divided into small, well defined steps, implemented as so called processors. Processors are grouped in modules, dedicated to particular tasks or types of analysis. Since many different groups worldwide are involved in the ILC project, it was assumed that the framework should allow distributed development of modules and should combine existing modules in a larger application according to users' needs. The crucial requirement in such an approach is that each step of the analysis has a well defined input and output data structure. In the case of Marlin, all possible data classes that can be exchanged between processors are defined in the LCIO (Linear Collider I/O) data model. LCIO is used by almost all groups involved in linear collider detector studies and thus has become a de facto standard in software development. By defining universal data structures we make sure that different processors can be connected in a single analysis chain, and can exchange data and analysis results.

The base class for a Marlin processor is also defined in the Marlin framework. It defines a set of standard callbacks that the user can implement in their subclasses. These callbacks are used to initialize the analysis chain, process subsequent sets of data and to conclude the analysis. A steering file mechanism allows the needed processors to be activated and their parameters to be set at run time. The dedicated processor manager loads selected processors and calls their corresponding methods for subsequent steps of data analysis. An example of the Marlin analysis chain for silicon pixel detectors, developed within the EUDET project [3], is shown in Fig. 3. Processing data from pixel detectors in High Energy Physics is in fact like CCD image analysis in astronomy. Charged particle tracks are measured instead of photons, but the analysis steps are similar: raw data are read from file, pixel cluster are found (object finding), their position and charge are reconstructed (photometry) and are used to fit the particle track (astrometry).

The Marlin framework has turned out to be very efficient and flexible, and is widely used by the ILC community. It is designed to run in a batch mode, without user interaction, with all input streams, analysis tasks, parameters and options specified in the steering file. This approach enables huge amounts of data to be handled and distributed computing resources to be used (Grid Computing). We therefore decided to use the same concept in developing the Luiza framework for GLORIA. The package is developed mostly in C++, and makes wide use of Standard Template Library (STL) classes and methods.

3 Luiza framework

3.1 Data structures

FITS[4] (Flexible Image Transport System) is the standard astronomical data format endorsed by both NASA and the IAU. FITS is much more than an image format (such as JPG or GIF), and is primarily designed to store scientific data sets consisting of multi-dimensional arrays (1-D spectra, 2-D images or 3-D data cubes) and 2-dimensional tables containing rows and columns of data. When developing Luiza, we decided to use CFITSIO[5] library for reading and writing data files in FITS format. The following basic data classes are defined:

The **GloriaFitsImage** class for storing 2-dimensional FITS images, with either integer or floating point pixels. It includes basic methods for image manipulation (addition, subtraction, multiplication and division);

The **GloriaFitsTable** class for storing data tables. Various column types are allowed (integers, floats, strings, and also vectors of integers and floats);

The **GloriaFitsHeader** class for storing FITS header data. This includes basic methods for accessing and modifying header information. Both GloriaFitsImage and GloriaFitsTable inherit from this class.

Additional classes can be implemented on this basis, for example the **GloriaObjectList** class. This class defines the GloriaFitsTable with predefined columns for storing object position on CCD ("CCD_X" and "CCD_Y") and object brightness ("Signal"). This ensures that object lists will be exchangeable between processors. A user can add additional columns if needed.

For internal storage of all data being processed, a dedicated class **GloriaDataContainer** was implemented. It stores vectors, so-called "collections", of images or tables. Each collection has a unique name (string), which can be used to access its elements. Multiple collections can be stored in memory, each with multiple images or tables (though in many cases collections will contain just a single image or table). The pointer to GloriaDataContainer is passed to each Luiza processor in the data processing loop. Processors can analyse data already stored in memory, and can also add new collections (e.g. when reading data from storage or saving analysis results).

3.2 Data processing

We assume that every computing task can be implemented as a processor (module) that analyzes the data stored in a GloriaDataContainer structure and additional output that is created is also added to that structure. A user defines the analysis chain at run time, by specifying a list of active processors in an XML steering file (see Fig. 1). The idea is to develope a large number of processors in GLORIA, performing many different tasks, so that the user is always able to find a set which matches his/her needs.

The main "work horse" of Luiza is the processor manager (ProcessorMgr class). This is used by Luiza to create a list of active processors (after parsing XML file), and to set values to the parameters required by these processors (given in the same XML file). The same processor type (e.g. a processor reading FITS images - FitsImageReader) can be used many times: the instances are distinguished by a unique names given by the user. Each instance has its own set of parameters, so one instance of an image reader can be used to read dark frames used for calibration, and another instance to read actual images; an example of a parameter section for one processor is shown in Fig. 4.

```
<execute>
  <processor name="DarkImageReader"/>
  <processor name="FlatImageReader"/>
  <processor name="RawImageReader"/>
  <processor name="CalibrateImage"/>
  <processor name="FindObjects"/>
  <processor name="DoAstrometry"/>
  <processor name="StoreFinalImage"/>
 </execute>
```

Figure 1: Example steering file header, containing information on the modules selected for the analysis chain.

3.3 Analysis tools

Until now we have mainly focused on the development of the general structure and functionality of Luiza: data classes based on the FITS standard have been designed, steer file parsing and processing management have been adopted from Marlin, processors for input and output of FITS image files have been implemented on the basis of CFITSIO library. Nevertheless, the current version of Luiza already includes some basic tools for image processing:

- an image viewer based on the CERN ROOT[6] package (see Fig. 5)
- an image normalization processor, allowing for dark/bias subtraction and flat correction
- a processor for image stacking or averaging
- a processor for simple geometry operations: image cropping and rotations
- two simple object finding algorithms: one based on the particle identification algorithm developed for silicon pixel detectors, and the other based on Python library Mahotas
- an astrometry algorithm based on Astrometry.net (still being tested)

In addition, a dedicated user interface, LuizaGUI, has been prepared for creating and editing of XML steering files.

3.4 Development plans

We plan to continue developing basic tools for image manipulation and analysis (astrometry, photometry, light curve reconstruction). General purpose algorithms, which should be flexible enough to cope with data from all GLORIA telescopes, are assumed not to be the most highly precise algorithms. They can be used as examples and starting points for future improvements, and in the development of more advanced tools, dedicated to particular studies. Dedicated processors will also be developed in the course of GLORIA off-line experiments. Our current plans include the development of:

- an interface to star catalogues and external databases
- an interface to Virtual Observatory resources
- a processor for smart image stacking (correcting for image shifts and rotations)
- frame quality analysis
- high quality aperture and profile photometries
- light curve determination and variability analysis
- searches for optical bursts, flares etc.

Thanks to the simple and modular structure of Luiza, individual GLORIA users will also be able to contribute to software development. New packages can be compiled as independent libraries and loaded dynamically at run time without the need to change anything in Luiza or other modules. It is therefore possible for users to develop "private" Luiza modules and libraries, adapted for their particular analysis, which can later be included in Luiza as a separate package after proper testing.

3.5 Documentation

We decided to use the Doxygen[7] package to manage framework documentation. Web page and/or LaTeX documentation is created automatically from class header files, based on simple tags used in the comments included in the code. The additional work needed to keep the documentation up to date is minimal, assuming that developers put comments in the code. This solution also makes it straightforward to add the code submitted by users to the documentation. Documentation for the public Luiza release is available on the dedicated web page[9].

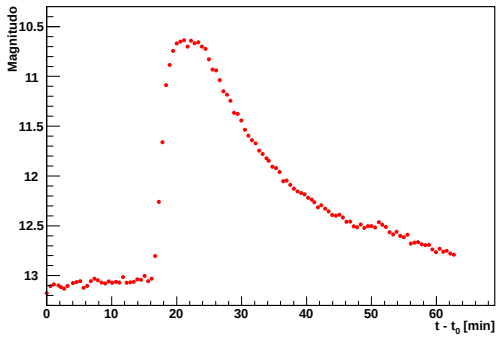

Figure 2: Lightcurve of flare star RXJ0413.4-0139 reconstructed with Luiza. Data from followup observation by Bootes-1 of the outburst observed by Pi of the Sky on Nov. 24, 2011. The secondary outburst of the star was measured.

4 Results

On November 24, 2011, just before midnight, the Pi of the Sky telescope located at INTA near Huleva, Spain, automatically recognized a new object in the sky. Unfortunately, it was visible to our detector only for about one minute, fading fast below our limiting magnitudo. We asked the Bootes group operating a bigger telescope at the same site to make a follow up observation. Figure 2 shows the light curve of the object, later identified as flare star RXJ0413.4-0139, as observed by the Bootes-1 telescope[8]. We clearly see a secondary outburst, more than one magnitudo brighter than the first one (11.8^m at maximum) and much, much longer. The data analysis resulting in this light curve was performed with Luiza.

5 Conclusions

An efficient and flexible analysis framework for GLORIA has been developed based on a concept taken from High Energy Physics. The basic data classes, framework structure and data processing functionality are implemented, as well as selected data processing algorithms. The framework will be further developed as a part of work on GLORIA off-line experiments. The first public version of the framework has been released via GLORIA project SVN[10].

Acknowledgement

The research leading to these results received funding from the European Commission Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 283783. The research work is also being funded the Polish Ministry of Science and Higher Education from 2011-2014, as a co-funded international project.

References

[1] http://www.gloria-project.eu/

[2] Gaede, F.: Marlin and LCCD: Software tools for the ILC, *Nucl.Instrum.Meth.* **A559**, 2006, 177-180. http://ilcsoft.desy.de/portal/software_packages/marlin/

[3] Rubinskiy, I.: EUTelescope final status, presented at EUDET Annual Meeting 2010, http://ilcagenda.linearcollider.org/conferenceDisplay.py?confId=4649

[4] http://fits.gsfc.nasa.gov/

[5] http://heasarc.gsfc.nasa.gov/docs/software/fitsio/fitsio.html

[6] http://root.cern.ch/

[7] http://www.doxygen.org/

[8] Martin Jelinek and Petr Kubanek, private communication.

[9] http://hep.fuw.edu.pl/u/zarnecki/gloria/luiza/doc/html/index.html

[10] http://sourceforge.net/projects/gloriaproject/

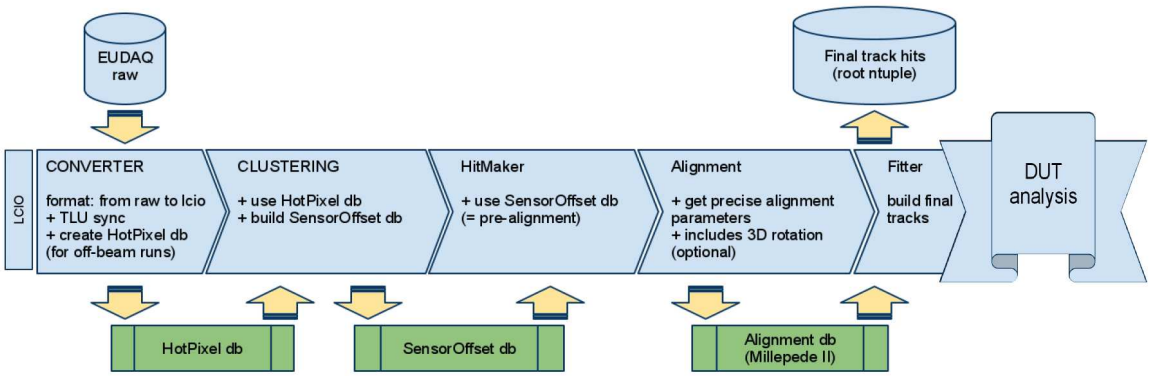

Figure 3: Example of the Marlin analysis chain for MIMOSA silicon pixel detectors, developed within the EUDET project (EUTelescope package).

```
<processor name="DarkImageReader" type="FitsImageReader">
  <!--Processor for reading input FITS images. Reads images from given files-->
  <!--List of FITS files to be read-->
  <parameter name="FitsFileList" type="StringVec"> dark.fit  </parameter>
  <!--Name of the image collection to which images from file should be stored-->
  <parameter name="ImageCollectionName" type="string">DarkImages </parameter>
  <!--Number of images to be read per processing loop (0 for all))-->
  <parameter name="ImagesPerLoop" type="int">0 </parameter>
  <!--Name of file containing FITS file list (one per line)-->
  <parameter name="ListFileName" type="string"> </parameter>
  <!--Flag for collections, which should not be deleted after loop is finished-->
  <parameter name="PermanentCollection" type="bool">true </parameter>
  <!--verbosity level for processor ("DEBUG,MESSAGE,WARNING,ERROR,SILENT")-->
  <parameter name="Verbosity" type="string"> MESSAGE </parameter>
</processor>
```

Figure 4: Example section of the steering file for Luiza, with parameters of the processor reading dark frames.

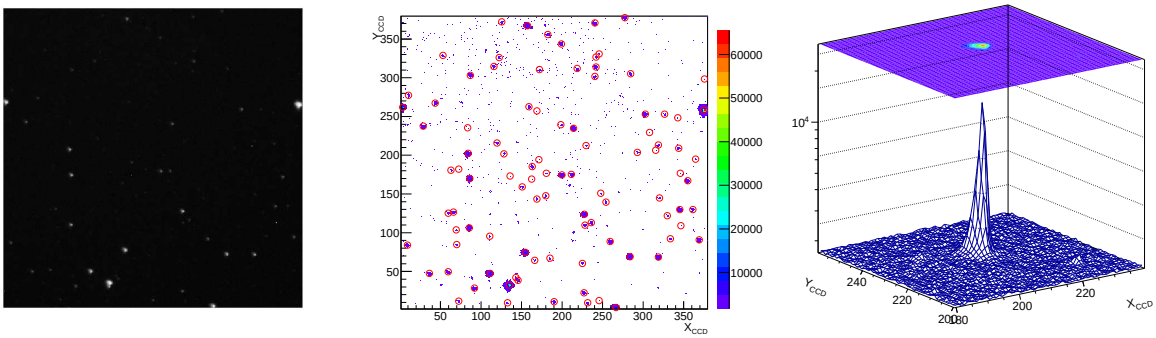

Figure 5: Various graphics options implemented in CERN ROOT, available for viewing FITS images in Luiza: as an image (left), as a histogram (centre), and as a histogram in 3D projection (right). Red circles in the middle plot indicate objects reconstructed in an image by the PixelClusterFinder processor. In the plot on the right, only small section of the image is shown for clarity, presenting the PSF of flare star RXJ0413.4-0139 at the outburst maximum.