Design, Implementation, and Optimization of MEGAlib's Image Reconstruction Tool Mimrec

Andreas Zoglauer[a,*], Steven E. Boggs[a], Michelle Galloway[a], Mark Amman[b], Paul N. Luke[b], R. Marc Kippen[c]

[a] *Space Sciences Laboratory, University of California at Berkeley, Berkeley, CA, USA*
[b] *Lawrence Berkeley National Laboratory, Berkeley, CA, USA*
[c] *Los Alamos National Laboratory, Los Alamos, NM, USA*

Abstract

MEGAlib, the Medium-Energy Gamma-ray Astronomy library, is a toolset to simulate and analyze data from gamma-ray detectors. An integral part of MEGAlib is its imaging tool Mimrec, which performs list-mode-likelihood image deconvolution. Mimrec has to handle data from coded masks, Compton cameras, and pair conversion telescopes with different response representations, on different imaging grids, with different deconvolution algorithms, etc. This versatility requires a highly modular and object-oriented design to avoid overhead and code redundancy. In addition, since some applications require close to real-time image reconstruction, great care has to be taken to optimize the library.

Keywords: MEGAlib, Mimrec, image reconstruction, data analysis, gamma-ray detector, Compton camera

1. Introduction

Gamma-ray detectors are utilized in many applications areas including astrophysics, nuclear medicine, and nuclear threat detection. The detectors implement a wide variety of imaging techniques such as coded masks, Compton scattering, pair conversion, and Laue diffraction. The underlying technology encompasses scintillators, time projection chambers, semiconductors (single volume, strip, and pixel detectors), as well as others. In addition, they can be arranged in a wide variety of geometries.

MEGAlib, the Medium-Energy Gamma-ray Astronomy library [1], is a tool for simulating and analyzing measured and simulated data of many of these instruments for terrestrial and space applications. It encompasses the complete data analysis chain: simulations, detector response creation, event reconstruction, performance assessments, and of course image reconstruction. The freely available, open-source software package has a completely object-oriented design, is written in C++, and is based upon ROOT [2].

The image reconstruction is performed by the program Mimrec (Megalib IMage REConstruction). A brief overview of its overall design, some implementation details, its optimization, and some examples are presented in the next sections.

2. Design and Implementation

The measurement process of any gamma-ray telescope can be described in the following way:

$$D(\vec{d}) = R(\vec{d}; \chi, \psi, E) \times J(\chi, \psi, E) + B(\vec{d}) \tag{1}$$

Photons emitted from an energy-dependent intensity distribution on the sky J (with χ, ψ as sky coordinates and E as energy) undergo the measurement process R and, together with some background B, are measured in the data space D with the parameters $\vec{d}$. The measurement process R completely describes the detector (absorption probabilities, detector noise, etc.) as well as all physical processes such as photo effect, Compton scattering, and pair creation.

The task of image reconstruction is to invert the measurement process and thus to determine the image $J^{rec}(\chi, \psi, E)$ from the given measurements $D(\vec{d})$ by utilizing the known response and background. Since no unique solution for this inversion problem exists, iterative approaches for the reconstruction of the image have to be chosen.

*Corresponding author. Address: Space Sciences Laboratory, 7 Gauss Way, Berkeley, CA 94720, USA. Email: zog@ssl.berkeley.edu. Tel: +1-510-643-7758. Fax: +1-510-643-8302

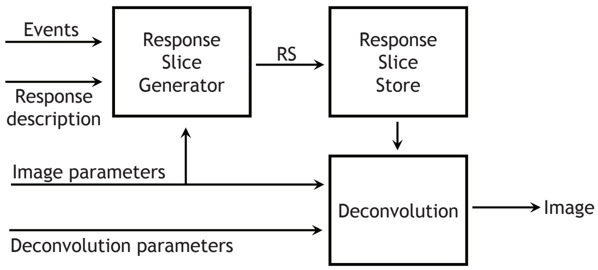

Figure 1: Basic layout of the imaging approach. See the text for details.

Mimrec implements list-mode-likelihood imaging methods such as the classic maximum-likelihood expectation-maximization approach [3]. In list mode, this method usually consists of two steps: first determine the response slices $R|_{\vec{d}_m}$ for all measured data sets $\vec{d}_m$, and then perform the image deconvolution. Figure 1 shows the basic layout of the implementation. Before determining the response slices, the correct response description is chosen based on the event type (e.g. Compton with or without electron track). Then, for the given grid, the response slice $R|_{\vec{d}_m}$ is determined or calculated based on the measured event data and grid position for all image bins. Next, the response slice is stored in an optimized way (e.g. sparse). When all response slices are determined, the deconvolution process is started using the response slices and some additional parameters. The result of the deconvolution is the reconstructed image. In Mimrec, the supervisor class MImager steers the image reconstruction and contains interfaces to set all parameters relevant for image reconstruction, which is usually done via a graphical user interface.

In order to be used in a wide variety of application scenarios, the image reconstruction approach has to be very versatile. Its most important design requirements include:

- Work for coded masks, Compton cameras with and without electron tracking, and pair-conversion telescopes in various instrument configurations
- Reconstruct images in Cartesian 2D and 3D as well as in astronomical coordinate systems
- Transparently handle different response representations
- Store the list-mode response slices in an optimized way to minimize RAM usage
- Transparently handle different deconvolution approaches
- Work close to real time for some application scenarios

In order to achieve the desired flexibility and to minimize code redundancy and overhead, a clear separation of all the different entities is required in the implementation: the data (e.g. the event types Compton and pair event), the response of the instrument, the calculated response slices, the image grid (e.g. Cartesian, spherical), and the deconvolution approaches.

In MEGAlib several different event types exist which are all derived from the abstract base class MPhysicalEvent. They are distinguished by the primary physical process the original photon has undergone in the detector: MComptonEvent, MPairEvent, MPhotoEffectEvent, MMuonEvent, and MUnknownEvent for an event which can not be identified. The base class MPhysicalEvent contains data which is common to all events such as ID, event timing, the telescope orientation in Galactic coordinates, etc. The derived classes contain all the specific event data and implement functions to calculate high level information such as scatter angles in the case of Compton events.

Determining the response slices is an integral part of the image reconstruction in list-mode. This involves iterating over the image grid. In order to incorporate different image-grid types (e.g. Cartesian 2D and 3D, astrophysical coordinate systems) different classes have been implemented. Their base class MBackprojection manages the basic grid information, the event data, the response, and the geometry. The derived classes such as MBackprojectionCartesian and MBackprojectionSpheric handle the looping over the different grid types in a way that is optimized for the specific grid.

The various detector response classes are designed in an similar way. The base class MResponse provides interfaces to retrieve the response in a common way. For example, for Compton events without electron track the MBackprojection classes determine the distance of the image bin center to the Compton cone in degrees. The classes derived from MResponse contain the Compton response parametrized accordingly, but with different attention to detail. For example, MResponseGaussian provides a simple Gaussian approximation of the Compton cone shape, and MResponsePBRM retrieves the shapes of the Compton cones as a function of the measured data from a large response matrix generated by simulations. This implementation allows for the easy enhancement of Mimrec with additional, detector specific response descriptions.

In order to calculate absorption probabilities, which are required to determine the geometry dependent part of the response, Mimrec uses Geomega (a description can be found in [4]), which is also part of MEGAlib. It enables the handling of a wide variety of detector types and instrument geometries.

After the response slice is determined, it has to be stored for later image deconvolution. Usually the amount of RAM necessary to store all these slices is one of the limiting factors for image reconstruction. Therefore several optimized storage options have been implemented. All these distinct implementations have the abstract base class MBPData in common. This base class provides a common interface for data storage and retrieval during the later deconvolution steps. The derived classes implement the concrete data storage in an optimized way. For example the class MBPDataImage just stores the data as a matrix of floats, and the class MBPDataSparseImageOneByte stores it as a sparse matrix where the content is encoded in 8 bits (i.e. 256 intensity levels) plus a floating-point number for the absolute normalization.

Finally, the class MLMLAlgorithms is the base class for all list-mode likelihood deconvolution algorithms, which manages the image parameters, response slice handling, stop conditions for the iterations, and all common data for the higher deconvolution algorithms. The derived classes, such as MLMLClassicEM and MLMLOSEM handle the specific deconvolution steps.

3. Optimization

Several application scenarios demand close to real-time imaging. One scenario is terrestrial nuclear threat detection with instruments such as HEMI, the High-Efficiency Multimode Imager, a combined Compton and coded-mask telescope [5]. Another scenario is gamma-ray burst localization aboard gamma-ray satellites such as the envisioned GRIPS telescope, a combined Compton scattering and pair conversion telescope [6].

Although most parts of Mimrec have been optimized, the following discussion is restricted to the response slice calculation for Compton telescopes, since this is the most challenging part.

In general, several possibilities exist to optimize the image reconstruction speed. An obvious one is parallelization on systems with multiple CPUs or cores. Mimrec is capable of performing the response slice calculation using multiple threads, each of which executes on a different CPU or core. For rather calculation intensive response descriptions, an almost linear scaling with number of threads is achieved. However, for simpler response descriptions, or when too many threads are used, the calculation is often limited by reading the events from file or by synchronizing the threads.

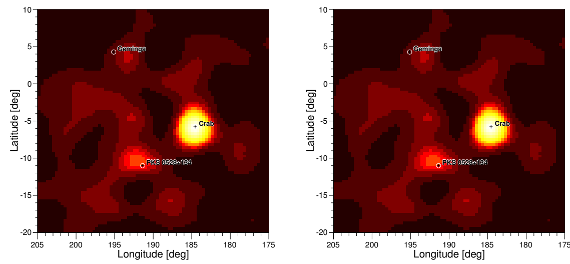

Figure 2: Galactic anti center with the pulsar Crab and the blazar PKS 0528+134, once calculated with the built-in mathematical function (left) and once using approximations (right)

Another option is the use of faster deconvolution algorithms. Besides others, Mimrec implements the ordered-subset maximum-likelihood expectation-maximization approach [7].

A further option would be to delegate some of the calculations to the graphics processing unit (GPU). However, since MEGAlib has to be widely portable and independent from any specific hardware and operating system, this approach has not been followed.

Moreover, finding the optimum compiler options usually also results in a performance improvement. Considering only the response slice calculation using an Intel Xeon Processor E5520 (2.27 GHz) on Ubuntu 9.10, the reconstruction speed is optimized by the following compiler options (here for gcc 4.3): ”-O3 -march=native -no-strict-aliasing”. This results in roughly 20% improvement compared to the standard ”-O2” option. Of course the improvement depends on the compiler version and the given CPU.

The most promising approach is, however, to optimize the code itself. A general approach is to use a profiler to uncover and eliminate bottlenecks in the code. With this approach, two critical areas can be identified: the used mathematical functions and the looping over the image grid.

The built-in mathematical functions provide much higher accuracy than needed for the final steps of image reconstruction (e.g. $\sim$16 digits for double). Although the built-in functions are highly optimized, replacing them with simpler approximations is advantageous. The most frequently used, time-intensive functions are acos(), exp(), and 1/sqrt(). While the first two are replaced by approximations found in [8], the latter is exchanged with the so called ”Quake-III” or ”Fast inverse square root” algorithm [9]. The achievable improvement varies with the used CPU, compiler, image-grid size, response, and application scenario. Figure 2 shows the Galactic anti center with the pulsar Crab and the blazar PKS 0528+134 as measured with COMPTEL [10], once calculated with the built-in mathematical functions (left) and once using approximations (right). While the speed improvement is 45%, the intensity difference is only 4×10^{-5}, which is negligible compared to an overall uncertainty of at least several percent.

Another important optimization is to accurately determine where to start the response calculation on the grid. Usually most of the bins on the image grid can be skipped, because the probability that the detected photon originated from these bins is below a certain threshold. For this task all response classes derived from MResponse provide a minimum and maximum angle around the measured Compton scatter angle, outside of which the response is below a certain threshold and thus can be neglected. In spherical coordinates the bins which fall within that range can be determined using spherical trigonometry. The performance improvements depend mostly on the detector response and the bin size, and range from 0% when the response covers the whole sky (because, e.g., the response includes incomplete absorption) to several 100% improvement for very narrow Compton cones. Furthermore, since the response slice is in many cases a sparse matrix, it should be handled from beginning to end as a sparse matrix to save further CPU cycles.

Finally, an overall performance test has been performed using simulations of the High-Efficiency Multimode Imager (HEMI). Typically, a large 1-m^2 instrument [11] measures a few thousand Compton events per second. Using a Gaussian approximation for the response of Compton events, and performing the image reconstruction on a 2-degree grid (angular resolution $\sim 5°$), results in a reconstruction performance of at least 7,500 Compton events per second and per CPU core for the used Intel Xeon E5520 CPU. This is more than enough for real-time imaging with HEMI.

For astrophysical real-time applications such as a gamma-ray burst mission such as GRIPS, the situation also appears favorable. Although the performance of space-grade computer systems is significantly below todays state-of-the-art hardware, only several tens of good (i.e. pre-selected) Compton events are necessary for the localization of a gamma-ray burst.

4. Summary

Mimrec is a versatile, highly-optimized image reconstruction tool. It can be applied to a wide variety of gamma-ray detectors on ground and in space and handles various event types (coded mask, Compton scattering, pair creation), on various image grids, using different response representations, and different deconvolution algorithms. It has been optimized to provide close to real-time imaging performance for some applications scenarios.

5. Acknowledgements

The design of Mimrec evolved over the course of the last decade. As consequence many different sources contributed to the funding of this work. Current funding originates from a LANL-IGPP University/Laboratory collaborative research project, from the U.S. Department of Homeland Security, Domestic Nuclear Detection Office, under Interagency Agreement HSHQDC-08-X-00832, and from the U.S. Department of Energy, Office of Science, under Contract DE-AC02-05CH11231.

References

[1] A. Zoglauer, R. Andritschke, F. Schopper, MEGAlib — The Medium Energy Gamma-ray Astronomy Library, New Astronomy Reviews 50 (2006) 629–632.

[2] R. Brun, F. Rademakers, ROOT - An Object Oriented Data Analysis Framework, NIM A 389 (1997) 81–86. See also http://root.cern.ch/.

[3] K. Lange, R. Carson, EM Reconstruction Algorithms for Emission and Transmission Tomography, Journal of Computer Assisted Tomography 8 (1984) 306–316.

[4] A. Zoglauer, G. Weidenspointner, S. E. Boggs, M. Galloway, C. B. Wunderer, Cosima - the Cosmic Simulator of MEGAlib, in: Nuclear Science Symposium Conference Record, 2009. NSS '09. IEEE.

[5] M. Amman, et al., Detector module development for the High Efficiency Multimode Imager, in: Nuclear Science Symposium Conference Record, 2009. NSS '09. IEEE, pp. 981–985.

[6] J. Greiner, et al., Gamma-ray burst investigation via polarimetry and spectroscopy (GRIPS), Experimental Astronomy 23 (2009) 91–120.

[7] M. Hudson, R. Larkin, Accelerated image reconstruction using ordered subsets of projection data, IEEE Transactions on Medical Imaging 13 (1994).

[8] M. Abramowitch, I. A. Stegun, Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables, Wiley, 1964.

[9] C. Lomont, Fast inverse square root, Technical Report, 2003. Available online: http://www.lomont.org/Math/Papers/2003/InvSqrt.pdf.

[10] V. Schönfelder, et al., Instrument description and performance of the imaging gamma-ray telescope COMPTEL aboard the Compton Gamma-Ray Observatory, The Astrophysical Journal Supplement Series 86 (1993) 657–692.

[11] A. Zoglauer, et al., First Results of the High Efficiency Multimode Imager, in: Nuclear Science Symposium Conference Record, 2009. NSS '09. IEEE.