

AIRY-LN: an ad-hoc numerical tool for deconvolution of images from the LBT instrument LINC-NIRVANA

Gabriele Desiderà^a, Andrea La Camera^a, Patrizia Boccacci^a, Mario Bertero^a, Marcel Carillet^b

^aDISI - Università di Genova, Via Dodecaneso 35, Genova, Italy

^bUMR 6525 H. Fizeau, Université de Nice Sophia Antipolis, CNRS, Observatoire de la Côte d'Azur, Campus Valrose, F-06108 Nice Cedex 2

ABSTRACT

LINC-NIRVANA (LN) is the German-Italian Fizeau beam combiner for the Large Binocular Telescope (LBT), composed of two 8.4-m apertures on a unique mount. It will provide multiple images of the same astrophysical target corresponding to different orientations of the 22.8-m maximum baseline. Starting from the already existing Software Package AIRY (a set of IDL-based modules developed within the CAOS “system” and dedicated to simulation and/or deconvolution of single or multiple images), an ad-hoc version has been especially designed for the data that will be obtained with LN. In this paper, we present the resulting Software Package AIRY-LN. Its capabilities, including quick-look methods, methods for specific classes of astronomical objects, PSF extraction, and a blind deconvolution algorithm are detailed. An IDL-licence-free (by means of the IDL Virtual Machine) and observer-oriented version of the whole package (with pre-setted LN image processing parameters) is also presented.

Keywords: Interferometric imaging - Data reduction - Blind deconvolution - PSF extraction - Large Binocular Telescope - LINC-NIRVANA - Software - AIRY-LN

1. INTRODUCTION

LINC-NIRVANA (LN)^{1,2} is the near-infrared image-plane beam combiner under construction for the Large Binocular Telescope (LBT)³ currently operating on Mount Graham in Arizona, with its two 8.4-m mirrors. Thanks to the Multi-Conjugate Adaptive Optics (MCAO) system, allowing for atmospheric turbulence compensation over a wide field of view, the performance of LN is expected to be very close to the diffraction limit. Due to the binocular nature of the instrument, its Point Spread Function (PSF) is similar to the PSF of an 8.4-m telescope crossed by the fringes due to the interference between the two apertures, characterized by a maximum baseline of approximately 22.8 m. For this reason, the raw images obtained with LN will have an anisotropic angular resolution: typical of an 8-m class telescope in one axis and approximately 3 times better in the orthogonal axis. The maximum resolution, in the direction of the maximum baseline, ranges from approximately 0.01 arcsec in band J to 0.02 arcsec in band K. If exposures are taken in different hour angles in order to obtain a better *uv*-coverage, an aperture-synthesis image reconstruction procedure can provide reconstructed images with the full angular resolution of a 22.8-m single-dish telescope.

Due to the next installation of the LN module on the telescope (scheduled for the end of 2010), a dedicated software for image reconstruction is needed. The Software Package AIRY-LN will be one of the collection of various numerical packages that will support the “Science” of LINC-NIRVANA. This software has been conceived to adapt and improve the already existing Software Package AIRY⁴, a CAOS-based⁵ general package for image simulations and reconstruction, publicly distributed (see <http://fizeau.unice.fr/caos> and <http://www.airyproject.eu>).

In addition, and as a part of the work around AIRY-LN, we are also studying new ad-hoc reconstruction algorithms (with regularized techniques and new accelerated methods), and performing a number of simulations of relevant scientific targets in order to test the quality of the algorithms as well the stability of the newly developed software package.

The main goal of AIRY-LN is to provide a flexible software to the LN community, able to solve various kinds of image reconstruction schemes by means of a collection of different algorithms, each one optimizing the restoration of a specific astronomical target.

2. THE SOFTWARE PACKAGE AIRY-LN

AIRY-LN has a modular structure (it consists of a set of specific modules, each representing a particular task), IDL-based, and designed to be used together with the CAOS Problem Solving Environment which provides a graphical interface, namely the CAOS Application Builder⁶. AIRY-LN, thanks to its structure, is very easy to modify. New features can be added creating a new module or modifying an old one.

The first version of AIRY-LN is based on a modification of the existent Software Package AIRY_4.0. Since AIRY-LN is planned to be used essentially for image reconstruction, all the AIRY modules concerning simulation were not considered while all the modules concerning image reconstruction have been optimized by taking into account the peculiar features of LN. The list of modules defined for AIRY-LN is shown in Fig.1. Most of them are adapted from AIRY_4.0, but module LDR (Ln De-Rotate image) and LTC (Ln data-Type Conversion). The acronyms being different from those of the AIRY modules, redundancy between the two packages is avoided and the user can use modules of the two packages in the same project without any confusion by the IDL interpreter. For example, one could use AIRY modules in order to simulate LN observations and then process the resulting data with the ad-hoc AIRY-LN modules. This is also possible because the CAOS System has been conceived in order to manage various packages together; in particular the module LTC of AIRY-LN is designed to convert the standard image data type of AIRY_4.0 (taken as input) to that of AIRY-LN.



Figure 1. List of modules composing AIRY-LN. Each of them admits from zero to two inputs or outputs, colored within the CAOS Application Builder as seen here (red if for the ad-hoc-defined “LN data” type and blue for the already existing “image” type used in AIRY_4.0).

This new Software Package not only consists in a copy of the old AIRY modules. A large number of modifications of the modules have been performed in order to be close to the peculiar LN features. Moreover it has been necessary to take into account the fact that the scientific camera of LN is fixed with the baseline, so that images acquired with different hour angles are rotated one with respect to each other. To compensate this feature a new module, which is able to perform image derotation, has been created. Moreover, it has been necessary to modify all the reconstruction algorithms in order to take into account the rotation of the object. Some examples of modifications/creations are the following:

- creation of new module LTC capable to join AIRY modules with AIRY-LN modules,
- creation of the new modules LPX and LDR able to handle up to 360 hour-angle images,
- new Graphical User Interfaces (GUIs) and deconvolution algorithms for modules LMD and LBD,
- implementation of new regularized reconstruction algorithms for LN images,
- extension of the pre-processing capabilities, including the handling of different numbers of frames per hour angle,
- improvement of the blind deconvolution algorithms,
- various modifications in routines inside different modules.

In addition the CAOS Application Builder itself has been modified in order to integrate the new type of input/output corresponding to the one used for LN data.

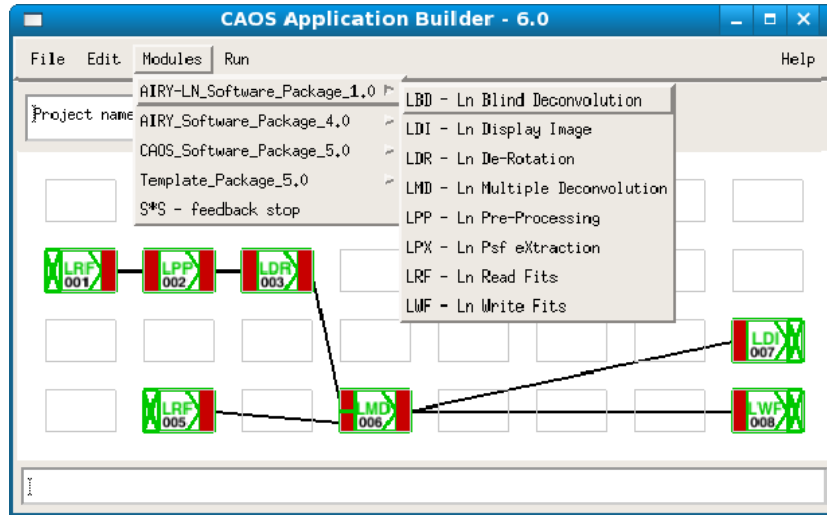


Figure 2. The CAOS Application Builder (an example of a deconvolution project is shown). From the pull-down menu **Modules**, first the list of packages installed by the user is shown. From this list, and selecting AIRY-LN, its modules are accessible.

2.1 AIRY-LN Project

The CAOS Application Builder interface allows the user to build a **reconstruction** program (**project**) by putting together the needed elementary modules and specifying the data flow between each block. When the project has been created the IDL code which implements the program is automatically generated. The user can set the physical and numerical parameters in the GUI characterizing each module.

In practice, each module is defined by a standard group of function calls, a collection of parameters and input(s) and output(s) of a pre-defined type. Each module can support up to two inputs and two outputs. As already mentioned, by clicking on the occurrence of a module, a GUI is opened in order to choose the various physical and numerical parameters related to the module itself. Global parameters (common to all modules) are limited to a strict minimum: only the total number of iterations and the current iterate number.

When the project design step is completed, it is possible to save the whole structure as a project that can be recovered for later modifications and/or parameters upgrading.

In the next sections we show some examples that represent various kinds of projects, namely **Deconvolution**, **Blind Deconvolution** and **PSF Extraction**.

Let us remark that AIRY-LN has been used (together with AIRY) to study the performance of LN for what concerns two specific kind of objects: a relatively distant galaxy at redshift ~ 1 and a Young Stellar Object (YSO) with its inner accretion disk region and collimated jet. A detailed analysis of these simulations is given in a paper presented in this Conference by Ciliegi et al.⁷.

3. RECONSTRUCTION ALGORITHM

A basic feature of the software is the use of a new approach for reconstruction algorithms. The idea is to implement a general function^{8,9} which can perform a simple reconstruction (OSEM¹⁰) or a regularized one by means of a different choice of its parameters. The implementation of this general algorithm will be a part of AIRY-LN, and thanks to its structure it can be used for applying a wide range of regularizations in the deconvolution process.

In Fig.2 we show an example of a **Project** performing a reconstruction by means of AIRY-LN modules. The data flow of the project is composed by:

- two modules (LRF) to read the image to be reconstructed and the PSF to be used,
- two modules to perform a **pre-processing** (LPP) and a **de-rotation** (LDI) of the images before performing the reconstruction,
- a **deconvolution module** (LMD),
- and finally the reconstructed images are saved (at each iteration or not, depending on what the user wants) using a module to **write a fits file format** (LWF).

Concerning the **pre-processing** module LPP, it will be able to perform a reduction of raw data by means of a technique that compensates the physical features of the detected images using **bad-pixel map**, **flat fields**, **dark fields**, **background estimation** and so on.

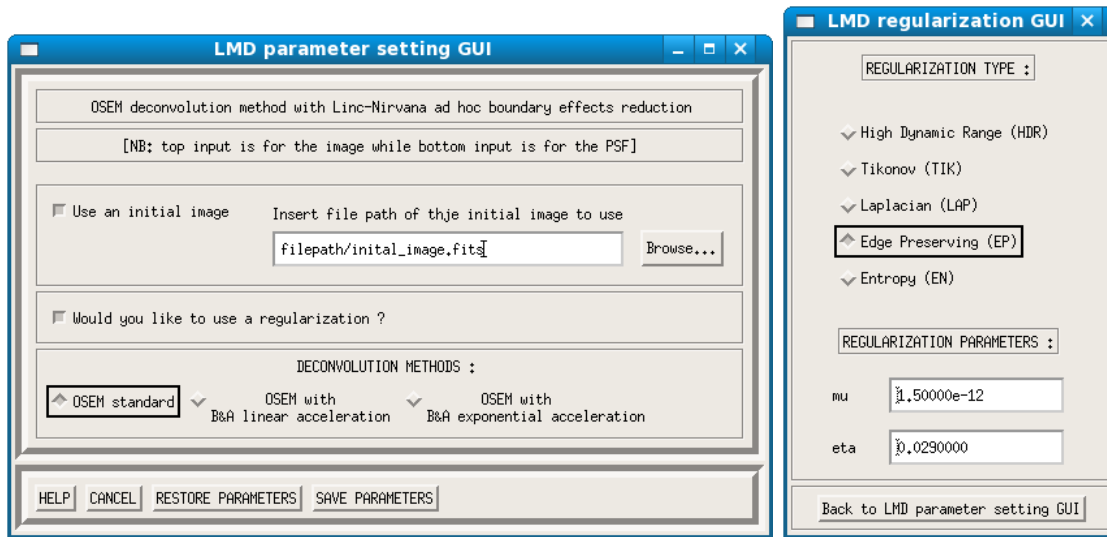


Figure 3. The GUIs for the LMD module (left) and the regularization (right).

Fig.3 shows the GUI of the module LMD together with the sub-GUI in which the parameters are set in the case a regularization is needed. As it can be seen, the parameters for the deconvolution are related to the initial image from which the reconstruction can start, the deconvolution method (standard OSEM or an accelerated version of it), and the regularization to be adopted if needed.

3.1 General implementation of the reconstruction algorithm

In this subsection we shortly describe the basic algorithm implemented in AIRY-LN. We denote by $g_j (j = 1, \dots, p)$ the images corresponding to the different hour angles and by f the object to be reconstructed. The starting problem without regularization is the minimization of the Kullback-Leibler divergence or Csiszár I-divergence¹¹

- minimize

$$J_0(f; g) = \sum_{j=1}^p \sum_{m \in S} \{g_j(m) \ln \frac{g_j(m)}{(A_j f)(m) + b_j(m)} + (A_j f)(m) + b_j(m) - g_j(m)\} \quad (1)$$

- subject to

$$f \geq 0 ; \quad \sum_{m \in R} \alpha(m) f(m) = \frac{1}{p} \sum_{j=1}^p \sum_{m \in S} \{g_j(m) - b_j(m)\} = c \quad (2)$$

where the array \bar{S} contains the image domain S (the intersection of the domains of the de-rotated images); R is the restoration domain (broader than the image domain S); A_j and A_j^T are defined in terms of the PSFs K_j by $A_j f = K_j * f$ and $A_j^T f = K_j^T * f$; finally $\alpha(m)$ is a weighting function, defined over \bar{S} as follows

$$\alpha(m) = \frac{1}{p} \sum_{j=1}^p \alpha_j(m) \quad , \quad \alpha_j(m) = \sum_{n \in S} K_j(n-m) \quad , \quad m \in \bar{S}.$$

With all the quantities introduced above and using the notation:

$$M_R(m) = \begin{cases} 1, & \text{if } m \in R, \\ 0, & \text{otherwise,} \end{cases}$$

the version of the algorithm implemented in LMD is:

- 1 Initialize the algorithm with $f^{(0)} = \text{const.}$, satisfying the flux condition of Eq.2.
- 2 For $k = 0, 1, 2, \dots$, given $f^{(k)}$, set $h^{(0)} = f^{(k)}$ and, for $j = 1, \dots, p$, compute

$$\begin{aligned} \tilde{h}^{(j)} &= \frac{M_R h^{(j-1)}}{\alpha_j} \left\{ A_j^T \frac{g_j}{A_j h^{(j-1)} + b_j} \right\} \quad , \\ \tilde{c}^{(j)} &= \sum_{m \in R} \alpha(m) \tilde{h}^{(j)}(m) \quad , \quad h^{(j)} = \frac{c}{\tilde{c}^{(j)}} \tilde{h}^{(j)}. \end{aligned}$$

- 3 Set $f^{(k+1)} = h^{(p)}$.

3.2 Regularizations

As just mentioned before an important topic is the use of AIRY-LN for investigating new penalty terms to create new regularized algorithms to optimize the reconstruction of images which are characterized by special features.

In the case of regularization, the functional to be minimized is:

$$J_\mu(f; g) = J_0(f; g) + \mu J_R(f). \quad (3)$$

As suggested in Lanteri et al.⁹, we consider the following decomposition of the gradient of the regularized functional Eq.3:

$$-\nabla J_R(f) = U_R(f) - V_R(f), \quad (4)$$

where U_R and V_R are suitable non negative arrays. By taking into account the penalized functional of Eq.3 the regularized algorithm will be:

- 1 Initialize the algorithm with $f^{(0)} = \text{const.}$, satisfying flux condition.
- 2 For $k = 0, 1, 2, \dots$, given $f^{(k)}$, set $h^{(0)} = f^{(k)}$ and, for $j = 1, \dots, p$, compute

$$\begin{aligned} \tilde{h}^{(j)} &= \frac{M_R h^{(j-1)}}{\alpha_j + \mu V_R(h^{(j-1)})} \left\{ A_j^T \frac{g_j}{A_j h^{(j-1)} + b_j} + \mu U_R(h^{(j-1)}) \right\} \quad , \\ \tilde{c}^{(j)} &= \sum_{n \in R} \alpha(n) \tilde{h}^{(j)}(n) \quad , \quad h^{(j)} = \frac{c}{\tilde{c}^{(j)}} \tilde{h}^{(j)}. \end{aligned}$$

- 3 Set $f^{(k+1)} = h^{(p)}$.

An implementation of a few regularizations, built with different penalty terms, are already available. These regularizations are obtained by means of the terms shown below from which we can derive the values of U_R and V_R by decomposing the gradient of the regularization functionals:

- High Dynamic Range

$$J_R(f) = \eta^2 \sum_n \frac{|f(n)|^2}{|f(n)|^2 + \eta^2} \quad ,$$

$$U_R(f) = 0 \quad ; \quad V_R(f) = \frac{2\eta^4 f}{(f^2 + \eta^2)^2}$$

- Tikhonov

$$J_R(f) = \frac{1}{2} \|f\|^2 \quad ,$$

$$U_R(f) = 0 \quad ; \quad V_R(f) = f$$

- Laplacian

$$J_R(f) = \frac{1}{2} \|(I - D)f\|^2 \quad ,$$

$$U_R(f) = (D + D^T)f \quad ; \quad V_R(f) = (I + D^T D)f$$

- Entropy

$$J_R(f) = \sum_n \left\{ f(n) \ln \frac{f(n)}{\bar{f}(n)} + \bar{f}(n) - f(n) \right\} \quad ,$$

$$U_R(f) = -\ln \frac{\bar{f}}{c} \quad ; \quad V_R(f) = -\ln \frac{f}{c}$$

- Edge-Preserving

$$J_R(f) = \frac{1}{4} \sum_n \sum_{n' \in \mathcal{N}(n)} \psi \left[\left(\frac{f(n) - f(n')}{\delta_{n,n'}} \right)^2 \right] \quad ,$$

$$U_R(f) = \sum_{n' \in \mathcal{N}(n)} \psi' \left[\left(\frac{f(n) - f(n')}{\delta_{n,n'}} \right)^2 \right] \frac{f(n')}{\delta_{n,n'}} \quad ; \quad V_R(f) = \sum_{n' \in \mathcal{N}(n)} \psi' \left[\left(\frac{f(n) - f(n')}{\delta_{n,n'}} \right)^2 \right] \frac{f(n)}{\delta_{n,n'}}$$

A detailed descriptions of all the regularizations shown above will be given in a paper, still in progress, about **Regularization methods for LBT LN image reconstruction**.

3.3 Acceleration techniques

All the algorithms studied until now have the common property to converge to the solution quite slowly; in general, all the regularized algorithms are slower than the standard algorithm. For this reason an acceleration technique is needed to improve the speed of the deconvolution process.

The acceleration techniques proposed by Biggs and Andrews^{12,13}, have been applied to the algorithms with a good decrease in terms of number of iterations to reach the same result provided by the standard algorithms without acceleration.

The chance to apply an acceleration is very important in the case of images provided by LN, because of two main reasons: the first one is the large dimension of the images (up to 2048×2048 pixels), while the second one is the presence of multiple images to reconstruct a single one.

A detailed description of the improvement due to the acceleration techniques adopted in AIRY-LN is described in a recent paper by La Camera et al.¹⁴

4. BLIND DECONVOLUTION

The implementation of a **Blind Deconvolution Algorithm** is very important for astronomical imaging (and more generally for an optimal astrophysical interpretation of the obtained data). In fact the PSF can be, in a number of common situations, too poorly known to permit a satisfactory deconvolution of the data. A very simple example is the case of extra-galactic objects with no suitable reference star in the image domain. For this reason an implementation of this technique must be optimized to improve the quality of the PSFs reconstruction together with the image to be restored.

A new idea concerning our implementation of blind deconvolution, is to use constraints on the reconstructed PSFs in order to force them to fit as far as possible the physical features estimated during the image acquisition.

The first new constraint, in addition to that on the **Band** already implemented in AIRY¹⁵, is applied on the **Strehl ratio**. The Strehl ratio value is nowadays used in modern optical astronomy in order to characterize the image quality that is obtained after adaptive-optics (AO) correction of the images. The higher the Strehl ratio, the closer to the ideal PSF. This new constraint has shown very encouraging results in the PSFs reconstruction within the framework of the preliminary tests we have performed. Details will be presented in a forthcoming paper.

The second idea of new constraint is related to the residual halo due to the AO system. A deeper study is definitively required and could lead to a new widely applicable deconvolution method.

Below are listed a **Project** and the GUIs in which the user can set all the parameters to perform a custom **Blind Deconvolution**. As just mentioned above, the **Blind Deconvolution** occurs when there is no decent PSF to be used for the reconstruction. In this case a theoretical initial PSF (or better, its autocorrelation) can be computed or the user can provide an estimation of it.

As concern the algorithms used inside the module LBD, OSEM (by now without any regularization) is adopted to reconstruct the images while Richardson-Lucy is used for each single PSF(s) orientation. Fig.4 shows an example of a project performing a blind deconvolution, while in Fig.5 and Fig.6 we show the GUIs in which set the parameters needed by the algorithms.

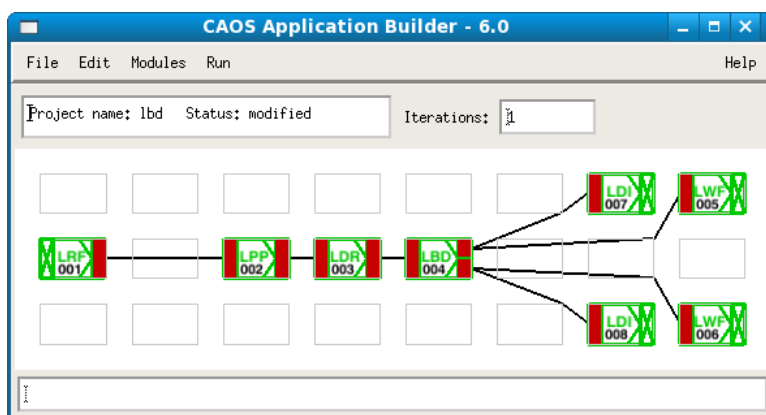


Figure 4. Example of a project performing a Blind Deconvolution.

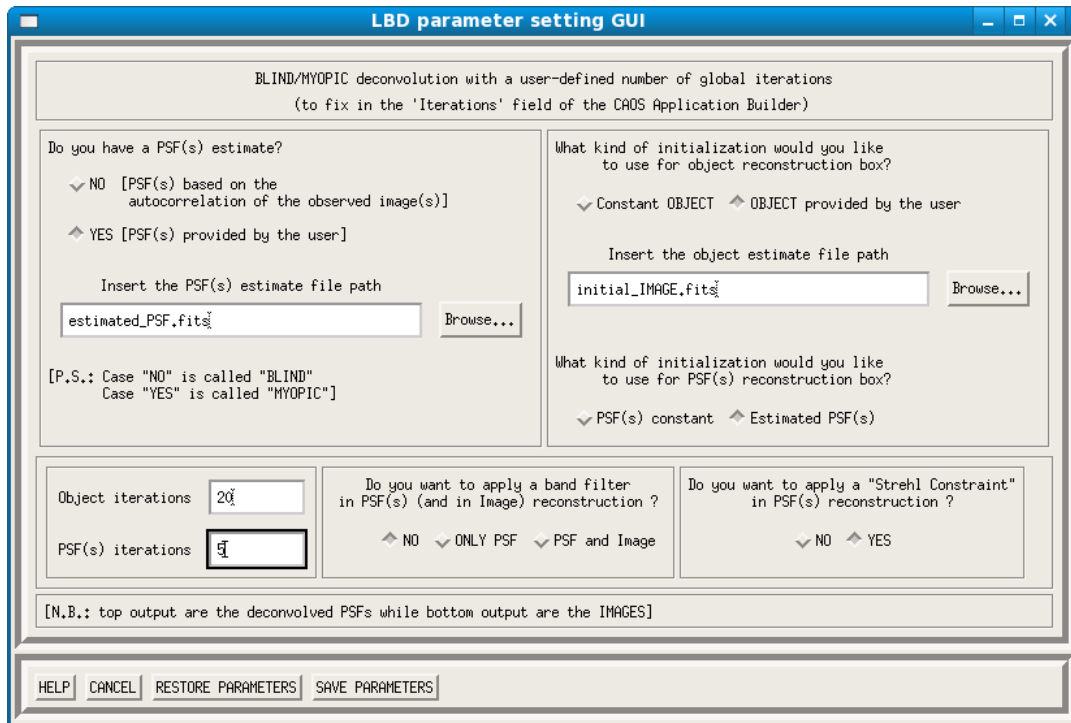


Figure 5. GUI in which parameters are set in order to initialize the Blind Deconvolution.

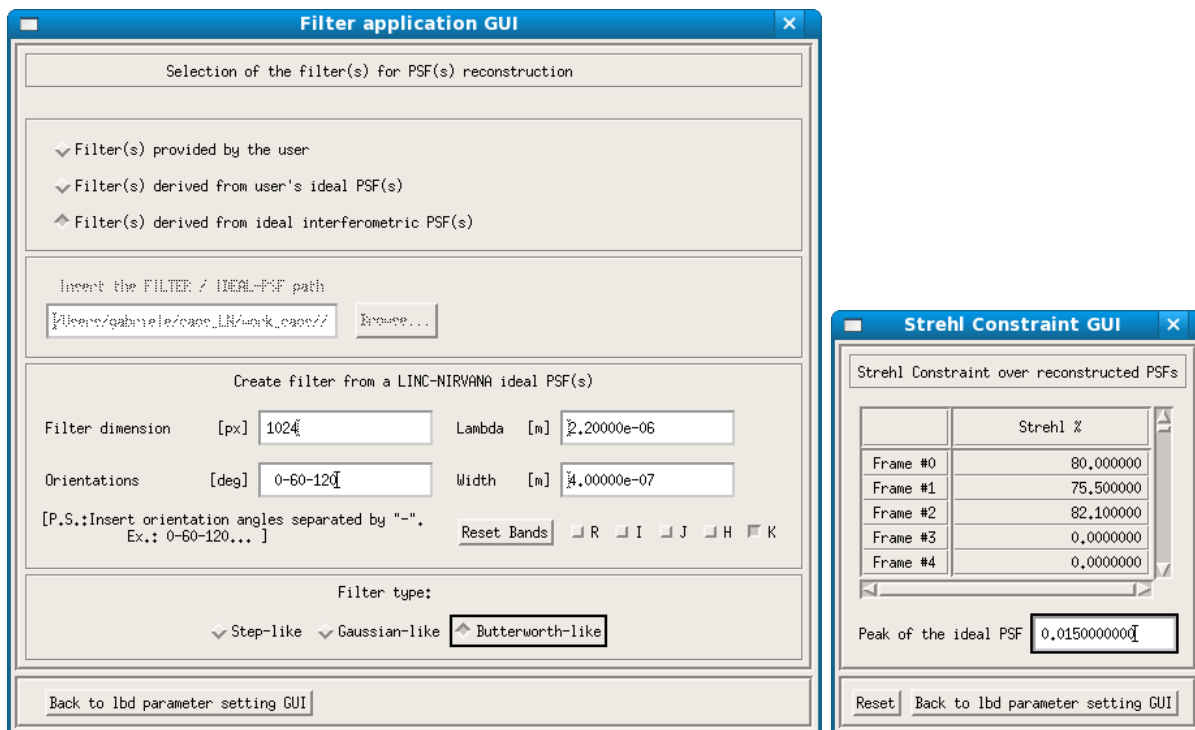


Figure 6. GUIs in which parameters are set in order to filter PSF(s)/image (left) and to set the Strehl constraint on the PSF(s) (right).

5. PSF EXTRACTION

The extraction of a PSF from an image is a necessary step in a deconvolution process. Indeed, the PSF is not known in the case of ground-based telescopes equipped with AO system as LBT, so it can be obtained from the image of a star contained in the field of view. In this section we describe a particular algorithm of “PSF extraction”. The method can be divided in two parts: extraction and extrapolation. The extraction part is performed by selecting a region around one or more stars. If the data represents multiple views of an object (one image per hour angle), the user must select one or more stars in each image.

The region has, generally, a smaller size than the input image. It could be immersed into a broader array by zero-padding, but this procedure is not acceptable for image deconvolution because it is equivalent to a sort of “masking” of the full PSF. In the uv plane, this is equivalent to blur the transfer function of the telescope, i.e. to convolve the full transfer function with the Fourier transform of the window function of the extraction domain. For this reason, we have implemented an extrapolation step.

The extracted region contains not only the image of the star but also a background (due to sky emission and/or to a nearby diffuse object). This background is estimated and then the image is fitted with a circular Lorentzian function. A more detailed description can be found in a paper by La Camera et al.¹⁴.

5.1 The LPX module

In this subsection we describe the Ln Psf eXtraction (LPX) module, available in AIRY-LN. It is derived from the Psf EXtraction (PEX) module of AIRY_4.0.

The module realizes the extraction and extrapolation algorithm described above in two different steps. The selection of star(s) is executed by the GUI, while the procedure is completed when the project is running (i.e. when the system executes the initializing program).

Obviously, in order to extract a PSF, it is necessary to load an input image. This operation can be performed by either the appropriate module (LRF) or inside the GUI of module LPX. The input of module LPX is hence optional and, depending on using it or not, one can define two “standard” PSF extraction projects as shown in Fig.7. In project A, the selection of the star used for PSF extraction is done during the setting up of the project, when the user chooses the parameters for each GUI. In fact, when the LPX GUI is opened, the user can load the image, select one or more stars and complete the other fields with necessary information. The position of the star(s) and the other parameters used in extraction/extrapolation step are now saved into a file that will be restored later, during the execution of the project.

In project B the LRF module loads the image and the normal execution flow is stopped until the LPX GUI is completed by the user. After this, the project can go on and the PSF is extracted as in project A. An advantage of project B is that, for example, a typical LN image could be derotated by module LDR before PSF extraction, by simply insert an occurrence of module LDR in between module LRF and module LPX.

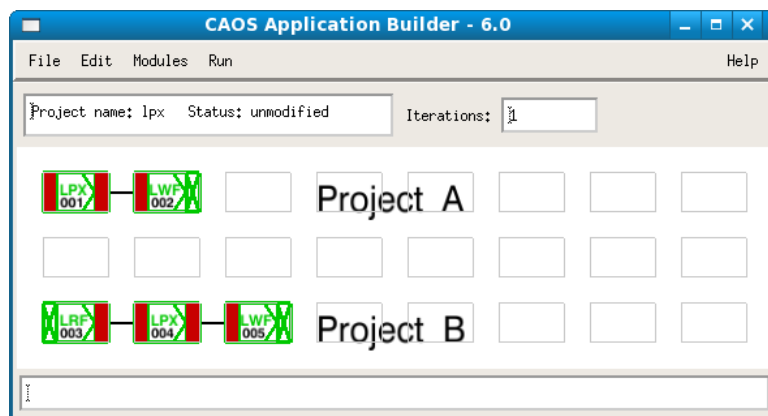


Figure 7. The two standard PSF extraction projects.

6. SIMULATION-RESTORATION PROJECT

In Fig.8 we show an example of a typical simulation-restoration project taking advantage from AIRY_4.1 (a forthcoming new version of the Software Package AIRY) and AIRY-LN modules. In the upper part of the project, AIRY_4.0 modules are used to simulate LN images, while in the bottom part AIRY-LN modules are used to compose a restoration project. The two sub-projects are joined each other by means of the LTC module.

The data flow of the whole project, can be shortly described with the following steps:

- SIMULATION (AIRY_4.1)
 - definition of the astronomical object (OBJ) and the PSF (RFT) to be used,
 - rotation of the object for each hour angle (RTI),
 - convolution (CNV) and subsequent noise perturbation (ADN),
- RESTORATION (AIRY-LN)
 - perform compatibility between the output of module ADN (from AIRY_4.1) and the input of module LPP (from AIRY-LN) input by means of module LTC,
 - pre-processing of the simulated data (LPP),
 - de-rotation of pre-processed data (LDR),
 - PSF extraction from de-rotated images (LPX),
 - Restoration process (LMD).

A detailed description of a more realistic simulation and the subsequent restoration is given in Ciliegi et al.⁷

7. VIRTUAL-MACHINE VERSION

Taking advantage from the Virtual Machine feature included in recent IDL versions (from 6.0), we are making different typical IDL-licence-free executables in order to perform the typical image reconstruction needed for astronomers observing with LN. The main parameters are settable while specialistic ones are pre-setted in order to face specific image reconstruction schemes.

This is practically an alternative version of AIRY-LN, which could be more directed to observers with a minimum of knowledge in image reconstruction techniques, while this Virtual-Machine-based version will be more useful for typical image reconstruction cases and/or non-expert observers.

Typical projects identified and being transformed this way are:

- image calibration,
- PSF extraction from a calibrated image,
- PSF extraction with on-the-fly image calibration,
- multiple deconvolution with calibrated PSF,
- blind deconvolution.

8. ACKNOWLEDGMENTS

This work was partially supported by INAF (Italian Institute for Astrophysics), through the project "Science with LINC-NIRVANA: the near-infrared interferometer for LBT", and by MUR (Italian Ministry for University and Research), Grant 2006018748.

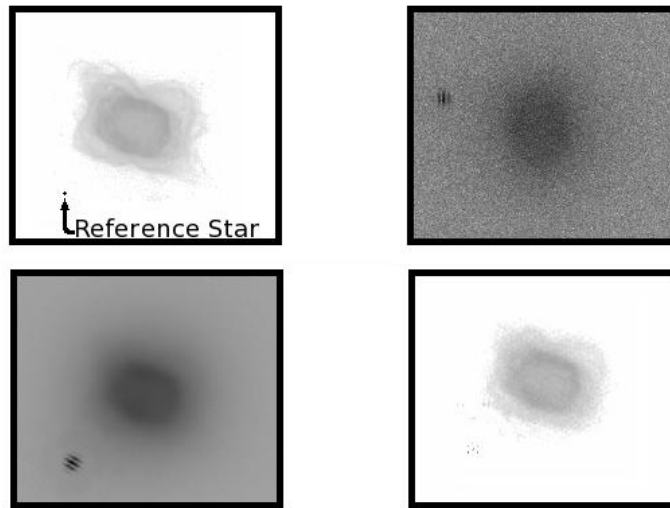
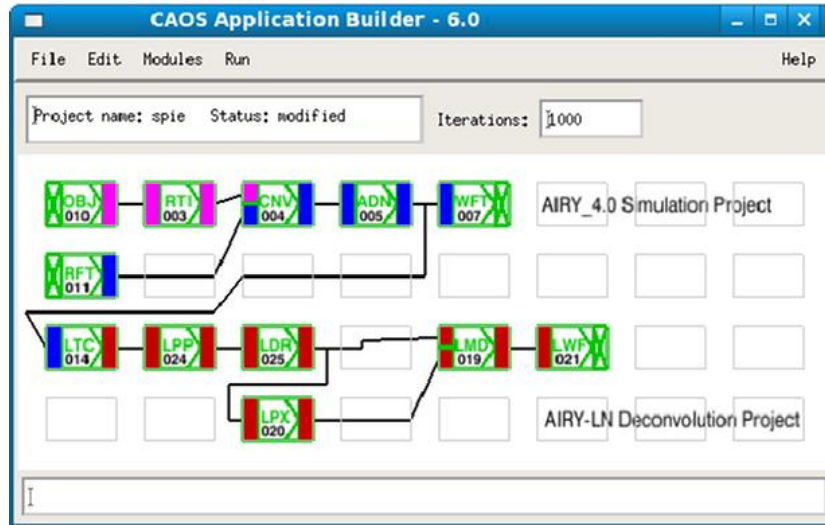


Figure 8. Upper panel: example of two projects linked by the module LTC in a unique one: the upper project is a simulation using modules from AIRY_4.1, while the lower one is a restoration using modules of AIRY-LN. In the bottom of the figure we show the unknown object (up-left), a single frame, of a set of 300, acquired for a specific hour angle (up-right), a de-rotated and calibrated image for the same hour angle (down-left) and the restored object (down-right).

REFERENCES

- [1] R.M.Wagner, “An overview of instrumentation for the large binocular telescope,” in [*Ground-based Instrumentation for Astronomy*], Iye, A. F. M. M., ed., *Proc. SPIE* **5492**, 108–120 (2004).
- [2] T.Herbst, R.Ragazzoni, A.Eckart, and G.Weigelt, “The linc-nirvana interferometric imager for the large binocular telescope,” in [*Ground-based Instrumentation for Astronomy*], Iye, A. F. M. M., ed., *Proc. SPIE* **5492**, 1045–1052 (2004).
- [3] P.Salinari, “The large binocular telescope interferometer,” in [*Optical Telescopes of Today and Tomorrow*],
- [4] S.Correia, M.Carbillet, P.Boccacci, M.Bertero, and L.Fini, “Restoration of interferometric images - 1. the software package airy,” *Astronomy & Astrophysics* **387**, 733–743 (2002).
- [5] M.Carbillet, C.Vérinaud, M.Guarracino, L.Fini, O.Lardière, B.LeRoux, T.Puglisi, B.Femenía, A.Riccardi, B.Anconelli, M.Bertero, and P.Boccacci, “Caos - a numerical simulation tool for astronomical adaptive

optics (and beyond),” in [*Advancements in Adaptive Optics*], D.Bonaccini, B. and R.Ragazzoni, eds., *Proc. SPIE* **5490(2)**, 550–559 (2004).

- [6] L.Fini, M.Carbillet, and A.Riccardi, “The caos application builder,” (2000). ADASS.
- [7] P.Ciliegi, A.LaCamera, G.Desiderà, S.Antoniucci, C.Arcidiacono, M.Lombini, E.Diolaiti, E.Bellocchi, F.Mannucci, M.Bertero, P.Boccacci, D.Lorenzetti, and B.Nisini, “Analysis of lbt linc-nirvana simulated images in two specific scientific cases,” in [*Optical and Infrared Interferometry*], *Proc. SPIE (this proceedings)* (2008).
- [8] H.Lanteri, M.Roche, O.Cuevas, and C.Aime, “A general method to devise maximum-likelihood signal restoration multiplicative algorithms with non-negativity constraints,” *Signal Processing* **81**, 945–974.
- [9] H.Lanteri, M.Roche, and C.Aime, “Penalized maximum likelihood image restoration with positivity constraint: multiplicative algorithms,” *Inverse Problems* .
- [10] M.Bertero and P.Boccacci, “Application of the os-em method to the restoration of lbt images,” *Astronomy & Astrophysics* **144**, 182–186 (2000).
- [11] I.Csiszár, “Why least squares and maximum entropy? an axiomatic approach to inference for linear inverse problems,” *The Annals of Statistics* **19 no.4**, 2032–2066 (1991).
- [12] D.S.C.Biggs and M.Andrews, “Acceleration of iterative image restoration algorithms,” *Applied Optics* **36 no.5**, 1766–1775 (1997).
- [13] D.S.C.Biggs and M.Andrews, “Multiframe iterative blind deconvolution using only a positivity constraint,” in [*Signal Recovery and Synthesis*], *OSA Technical Digest Series* **11**, 8–10 (1998).
- [14] A.LaCamera, G.Desiderà, C.Arcidiacono, P.Boccacci, and M.Bertero, “Advances in the reconstruction of lbt linc-nirvana images,” *Astronomy & Astrophysics* **471**, 1091–1097 (2007).
- [15] G.Desiderà, B.Anconelli, M.Bertero, P.Boccacci, and M.Carbillet, “Application of iterative blind deconvolution to the reconstruction of lbt linc-nirvana images,” *Astronomy & Astrophysics* **452**, 727–743 (2006).