

AIRY: a complete tool for the simulation and the reconstruction of astronomical images

Andrea La Camera^a, Marcel Carillet^b, Chiara Olivieri^a, Patrizia Boccacci^a, Mario Bertero^a

^aDipartimento interscuola di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi (DIBRIS), Università di Genova, Via all'Opera Pia 13, 16145, Genova, Italy;

^bUMR 7293 Lagrange, Université de Nice Sophia Antipolis/CNRS/Observatoire de la Côte d'Azur, Parc Valrose, 06108 Nice Cedex 2, France

ABSTRACT

The **Software Package AIRY** (acronym for **Astronomical Image Restoration in interferometrY**) is a software tool designed to perform simulation and/or deconvolution of images of Fizeau interferometers as well as of any kind of optical telescopes. AIRY is written in IDL and is a Software Package of the **CAOS Problem Solving Environment** (PSE): it is made of a set of modules, each one representing a specific task. We present here the last version of the software, arrived at its sixth release after 10 years of development. This version of AIRY summarizes the work done in recent years by our group, both on AIRY and on AIRY-LN, the version of the software dedicated to the image restoration of LINC-NIRVANA (LN), the Fizeau interferometer of the Large Binocular Telescope (LBT). AIRY v.6.0 includes a renewed deconvolution module implementing regularizations, accelerations, and stopping criteria of standard algorithms, such as OSEM and Richardson-Lucy. Several modules of AIRY have been improved and, in particular, the one used for the extraction and extrapolation of the PSF. In addition, AIRY has modules dedicated to the simulation of interferometric images and utility modules for data reading, writing, and visualization. After a description of the implemented reconstruction methods and of the whole set of modules, we provide several example projects in order to give to the astronomical community a powerful tool for the preparation of the observations and for the real data deconvolution.

Keywords: Software Package AIRY, CAOS Problem Solving Environment, image deconvolution, image denoising, PSF extraction.

1. INTRODUCTION

The **Software Package AIRY** (**Astronomical Image Reconstruction in interferometrY**) was born at the beginning of this millennium^{1,2} as a set of modules for the simulation and/or analysis of a set of Fizeau interferometric images taken at different orientations of the baseline, including a deconvolution module implementing the OSEM (ordered subset expectation maximization) method proposed by Bertero & Boccacci^{3,4} (OSEM being an accelerated version of the extension of the Richardson-Lucy (RL) method to multiple-images deconvolution), and following a preliminary work on the OSEM-based deconvolution of simulated post-adaptive optics images from the Large Binocular Telescope (LBT), presented at the astronomical SPIE meeting of year 2000 in Munich, Germany⁵.

Although the privileged field of application of the various methods finally implemented within the package was the Fizeau interferometric beam combiner LINC-NIRVANA of the LBT (see e.g. the reviews of Bertero et al.^{6,7}), it rapidly presented a general approach which can be used for designing a wide class of deconvolution methods including not only OSEM and accelerated versions of OSEM but also methods for Bayesian approaches to the restoration of specific classes of objects, with application either for the case of Fizeau interferometric multiple images, but also standard mono-pupil data such as post-adaptive optics images from any monolithic optical telescope, and even mono dimensional data from a spectrograph aboard them.

Further author information: (Send correspondence to A.L.C., M.C.)

A.L.C.: E-mail: andrea.lacamera@unige.it, Telephone: +39 010 353 6981

M.C.: E-mail: marcel.carillet@unice.fr, Telephone: +33 492076328

From the beginning, the **Software Package AIRY** has been developed within the **CAOS problem solving environment (PSE)** (see Carillet et al.^{8,9} and <http://lagrange.oca.eu/caos>) as a companion package to the original **Software Package CAOS**¹⁰ dedicated to the modeling of adaptive optics (AO) systems. As for the other public parts of the **CAOS PSE**, the package is freely distributed among the astronomical community, and both a mailing-list and a specific website (<http://www.airyproject.eu>) are dedicated to it.

After the first *A&A* paper² presenting the package in 2002, and a companion paper on the study of the interferometric imaging capabilities of LBT with respect to partial AO correction and partial angular coverage¹¹, a series of papers developing new methods followed, including computationally efficient methods for a somehow sophisticated “quick look” of interferometric Fizeau data¹², super-resolution through deconvolution¹³, reduction of the boundary effects for wide field-of-view images¹⁴, deconvolution of images with a high dynamic range^{15,16}, iterative blind deconvolution (IBD) for LBT images¹⁷, and the implementation of a Strehl constraint for IBD of post-adaptive optics data in general¹⁸. All the developments cited were implemented within the package, which was also used in practice for non-interferometric real data^{19–21}.

Since 2007 we are also developing the **Software Package AIRY-LN**, specifically dedicated to the reconstruction of LN images^{22,23}. Obviously the experience we developed within **AIRY-LN** permitted us to improve **AIRY** and vice-versa.

The paper is organized as follows. In Sect. 2 we describe the reconstruction methods available, while in Sect. 3 we give a summary of the present status of the whole package and a description of the new features implemented in the forthcoming version of **AIRY**. Then, in Sect. 4 we describe some examples of use and results obtainable from our numerical experiments. Finally Sect. 5 consists of some perspectives for upcoming developments of **AIRY**.

2. THE RECONSTRUCTION METHODS

In several papers^{4,6,24} we proposed the general approach to the design of reconstruction methods, and in particular in a recent paper⁷ we discussed the image restoration problem thoroughly. Since most of the algorithms proposed in our recent papers have been implemented only in the **Software Package AIRY-LN**, and since we need to update the existent reconstruction algorithms present in the **Software Package AIRY**, we summarize here the model of image formation and the deconvolution problem.

2.1 The model of image formation

We denote by $\mathbf{f} = \mathbf{f}(\mathbf{n})$ the unknown object to be imaged and by $\mathbf{g} = \mathbf{g}(\mathbf{m})$ the detected image. Moreover, we denote by $\mathbf{n} = \{n_1, n_2\} \in R$ the multi-index of the pixels in the object domain R and by $\mathbf{m} \in S$ the multi-index of the pixels in the image domain S . Generally speaking, the two sets R and S may not coincide and we suppose that R is broader than S . This situation occurs if the object has an angular size greater than the Field of View (FoV) of the instrument. We also suppose to merge the two sets into a broader array \tilde{S} by surrounding the arrays with zeros (zero-padding) such as $S \subset R \subset \tilde{S}$. We again call \mathbf{f} and \mathbf{g} the object and the image, respectively, defined in \tilde{S} . In the case of multiple-images, we denote by \mathbf{g} the set of the p images \mathbf{g}_j , with $j = 1, \dots, p$.

The model we use of image formation of each image \mathbf{g}_j was introduced by Snyder et al.²⁵ and is written:

$$\mathbf{g}_j(\mathbf{m}) = \mathbf{g}_j(\mathbf{m})^{(\text{obj})} + \mathbf{g}_j(\mathbf{m})^{(\text{back})} + \mathbf{g}_j(\mathbf{m})^{(\text{ron})} \quad , \quad (1)$$

where $\mathbf{g}_j(\mathbf{m})^{(\text{obj})}$ is the number of photoelectrons arising from the object radiation; $\mathbf{g}_j(\mathbf{m})^{(\text{back})}$ is the number of background photoelectrons (including sky background, dark current, and bias) and $\mathbf{g}_j(\mathbf{m})^{(\text{ron})}$ is the amplifier read-out noise. The first two terms are described by a Poisson process, the first with expected value $(\mathbf{A}_j \mathbf{f})(\mathbf{m})$ and the latter with expected value $\mathbf{b}_j(\mathbf{m})$. The last term is described by an additive Gaussian process, with expected value μ and variance σ^2 . Without changing the statistics, we assume $\mu = 0$ for simplicity.

We denote by \mathbf{A}_j the matrix describing the effect of the optical components and the atmosphere. In most applications it can be a Toeplitz or a circulant matrix, i. e. $\mathbf{A}_j(\mathbf{m}, \mathbf{n}) = \mathbf{A}_j(\mathbf{m} - \mathbf{n})$. Moreover, in the case of space invariant images we can compute the term $\mathbf{A}_j \mathbf{f}$ with convolution and more precisely (by indicating with

\mathbf{K}_j the j -th PSF) we can write $\mathbf{A}_j \mathbf{f} = \mathbf{K}_j * \mathbf{f}$. Finally we suppose, as usual, that the PSFs are normalized to a unit volume:

$$\sum_{\mathbf{n} \in R} \mathbf{K}_j(\mathbf{n}) = 1, \quad j = 1, \dots, p. \quad (2)$$

We introduce the following operations between a generic array \mathbf{f} in the object domain and a generic array \mathbf{g} in the image domain:

$$(\mathbf{A}_j \mathbf{f})(\mathbf{m}) = \sum_{\mathbf{n} \in R} \mathbf{K}_j(\mathbf{m} - \mathbf{n}) \mathbf{f}(\mathbf{n}), \quad \mathbf{m} \in S ; \quad (3)$$

$$(\mathbf{A}_j^T \mathbf{g})(\mathbf{n}) = \sum_{\mathbf{m} \in S} \mathbf{K}_j(\mathbf{m} - \mathbf{n}) \mathbf{g}(\mathbf{m}), \quad \mathbf{n} \in R . \quad (4)$$

2.2 Deconvolution

The image restoration problem requires an accurate statistical model of the images. As said above, in each pixel we have the sum of three terms, two of them are realizations of independent Poisson random variables and the third of an additive Gaussian process due to Read-Out-Noise (RON). In this way the noise is a mixture of Gaussian and Poisson noise. We discussed the complete model of noise in Benvenuto et al.²⁶ However, in most cases (for example in infrared astronomical images with a large background), a sufficiently accurate approximation is obtained by adding the variance σ^2 of the RON to the images and to the background values, so that they can be modeled as independent Poisson processes²⁷.

In other cases²⁸, the noise can be modeled as a Gaussian process. In the following we cope with both *only Poisson* and *only Gaussian* noise, remembering that, in the case of Gaussian+Poisson, we can reduce the model to the case *only Poisson* noise simply by adding the term σ^2 to the images and to the background.

The general approach to the deconvolution methods follows from the minimization of a functional of the type $J_\mu(\mathbf{f}; \mathbf{g}) = J_0(\mathbf{f}; \mathbf{g}) + \mu J_1(\mathbf{f})$, with the additional constraint $\mathbf{f} \geq 0$. The minimizers of J_0 in general do not provide a reliable solution of the reconstruction problem and, for this reason, solutions can be obtained by means of a Bayesian approach with the minimization of the functional J_μ . The functional J_0 measures the discrepancy between the computed images associated to \mathbf{f} and the detected images \mathbf{g} and is called *data-fidelity function*. According to the noise model it can be described either by the sum of generalized Kullback-Leibler divergences (also called Csizár I-divergence²⁹), one for each image, or by the sum of least-square functionals:

$$J_0(\mathbf{f}; \mathbf{g}) = \sum_{j=1}^p \sum_{\mathbf{m} \in S} \left\{ \mathbf{g}_j(\mathbf{m}) \log \frac{\mathbf{g}_j(\mathbf{m})}{(\mathbf{A}_j \mathbf{f} + \mathbf{b}_j)(\mathbf{m})} + (\mathbf{A}_j \mathbf{f} + \mathbf{b}_j)(\mathbf{m}) - \mathbf{g}_j(\mathbf{m}) \right\} \quad \text{Poisson noise;} \quad (5)$$

$$J_0(\mathbf{f}; \mathbf{g}) = \frac{1}{2} \sum_{j=1}^p \|\mathbf{A}_j \mathbf{f} + \mathbf{b}_j - \mathbf{g}_j\|^2 \quad \text{Gaussian noise.} \quad (6)$$

The functional J_1 is called *regularization function* and the parameter $\mu > 0$ is called *regularization parameter*. In Bertero et al.⁷ we investigated several regularization functions and in Tab. 1 we give their expressions. In the table, with respect to the two first regularizations, the euclidean norm is denoted by $\|\cdot\|$, I is the constant array with 1 in all entries, while D can be obtained from the 3×3 mask with zero value in the center and in the four corners and $1/4$ in the other positions. Concerning the edge-preserving regularization, $\mathcal{N}(\mathbf{n})$ is the set of the eight pixels surrounding the pixel \mathbf{n} , $\delta_{\mathbf{n}, \mathbf{n}'} = 1$ for horizontal/vertical neighbours and $\delta_{\mathbf{n}, \mathbf{n}'} = \sqrt{2}$ for diagonal neighbours, and $\psi(t) = 2\sqrt{t + \eta^2} - \eta$ where η is a thresholding parameter. In the case of entropy regularization, the functional J_1 is the Kullback-Leibler divergence between the object and a nonnegative reference array $\bar{\mathbf{f}}$. The last regularization (HDR) is a sort of Tikhonov regularization with a thresholding parameter η . Further details on HDR regularization can be found in Anconelli et al.¹⁵.

These functionals have well-known good properties, so that absolute minimizer exists. In order to minimize the functional J_μ we can follow the Split Gradient Method (SGM) proposed by Lantéri et al.³⁰ The basic idea is to decompose the gradient of each functional in two non-negative functions:

$$-\nabla J_0(\mathbf{f}; \mathbf{g}) = U_0(\mathbf{f}; \mathbf{g}) - V_0(\mathbf{f}; \mathbf{g}) ; \quad -\nabla J_1(\mathbf{f}) = U_1(\mathbf{f}) - V_1(\mathbf{f}) . \quad (7)$$

name	J_1
Tikhonov	$\frac{1}{2} \ \mathbf{f}\ ^2$
Laplacian	$\frac{1}{2} \ (I - D)\mathbf{f}\ ^2$
Edge-preserving	$\frac{1}{4} \sum_{\mathbf{n} \in R} \sum_{\mathbf{n}' \in \mathcal{N}(\mathbf{n})} \psi \left[\left(\frac{\mathbf{f}(\mathbf{n}) - \mathbf{f}(\mathbf{n}')}{\delta_{\mathbf{n}, \mathbf{n}'}} \right)^2 \right]$
Entropy	$\sum_{\mathbf{n} \in R} \mathbf{f}(\mathbf{n}) \log \frac{\mathbf{f}(\mathbf{n})}{\bar{\mathbf{f}}(\mathbf{n})} + \bar{\mathbf{f}}(\mathbf{n}) - \mathbf{f}(\mathbf{n})$
High dynamic range	$\frac{\eta^2}{2} \sum_{\mathbf{n} \in R} \frac{ \mathbf{f}(\mathbf{n}) ^2}{ \mathbf{f}(\mathbf{n}) ^2 + \eta^2}$

Table 1. The regularization functions implemented in AIRY.

The minimizer \mathbf{f}^* satisfies the first Karush-Kuhn-Tucker (KKT) condition, i.e. $\mathbf{f}^* \nabla J_\mu(\mathbf{f}^*; \mathbf{g}) = 0$. The condition leads to a fixed-point equation and, by applying the method of successive approximations, we can obtain the general algorithm:

- initialize the algorithm with $\mathbf{f}^{(0)} > 0$;
- for $k=0,1,2,\dots$ given $\mathbf{f}^{(k)}$ compute:

$$\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} \frac{U_0(\mathbf{f}^{(k)}; \mathbf{g}) + \mu U_1(\mathbf{f}^{(k)})}{V_0(\mathbf{f}^{(k)}; \mathbf{g}) + \mu V_1(\mathbf{f}^{(k)})}.$$

In the following table we give the expression of U_0 and V_0 in the case of Poisson and Gaussian noise. The first row gives the well-known Richardson-Lucy^{31,32} algorithm in the case of multiple images, while the second row gives the image space reconstruction algorithm (ISRA)³³ used in the case of Gaussian noise.

noise	U_0	V_0
(P)	$\sum_{j=1}^p \mathbf{A}_j^T \frac{\mathbf{g}_j}{\mathbf{A}_j \mathbf{f}^{(k)} + \mathbf{b}_j}$	$\boldsymbol{\alpha} = \sum_{j=1}^p \boldsymbol{\alpha}_j$
(G)	$\sum_{j=1}^p \mathbf{A}_j^T \mathbf{g}_j$	$\sum_{j=1}^p \mathbf{A}_j^T \mathbf{A}_j \mathbf{f}^{(k)} + \mathbf{b}_j$

Table 2. Standard algorithms implemented in AIRY. The definition of the arrays $\boldsymbol{\alpha}_j$ is given in Eq. 8.

In general, as remarked at the beginning of this section, the object is reconstructed in the broader array R , if it is convenient to apply the correction of boundary effects^{14,34}. In this case, as concerns the RL algorithm, each array $\boldsymbol{\alpha}_j$ is defined by:

$$\boldsymbol{\alpha}_j(\mathbf{n}) = \sum_{\mathbf{m} \in S} \mathbf{K}_j(\mathbf{m} - \mathbf{n}) \chi_S(\mathbf{m}), \quad \mathbf{n} \in R, \quad (8)$$

where χ_S is the characteristic function of S in R and we use equation (4) to compute the last term. In other words, this is a sort of “convolution” with the mask χ_S . In the case of $S = R = \bar{S}$, i.e. if the correction of boundary effects is not necessary, each $\boldsymbol{\alpha}_j$ is simply the constant array with 1 in all entries, thanks to the normalization of the PSF defined in equation (2) and periodic continuation of the PSF. In the case of ISRA, we remark that, in the term V_0 , the operation $\mathbf{A}_j^T \mathbf{A}_j \mathbf{f}^{(k)}$ must be computed in two separate steps, because we need

to apply the mask χ_S to the result of $\mathbf{A}_j \mathbf{f}^{(k)}$ before computing the external operation. Again, in the case of $S = R = \bar{S}$ (no boundary correction) this operation is not necessary and $\mathbf{A}_j^T \mathbf{A}_j \mathbf{f}^{(k)}$ can be efficiently computed by noting that, in the Fourier domain, it corresponds to $|\widehat{\mathbf{K}}_j|^2 \cdot \widehat{\mathbf{f}}^{(k)}$.

We also consider a block decomposition of the gradient that suggests the technique of *ordered subset* (OS), proposed for Emission Tomography³⁵. In this way each image can be considered as a projection in 2D tomography. The two functions U_0 and V_0 can be written as the sum of p functions U_{0j} and V_{0j} . Both the RL and ISRA algorithm have an OS version: the first is the well-known OSEM algorithm³⁶, while the latter is called OS-ISRA. Both methods lead to the following algorithm:

- initialize the algorithm with $\mathbf{f}^{(0)} > 0$;
- for $k=0,1,2,\dots$ given $\mathbf{f}^{(k)}$, set $\mathbf{h}^{(0)} = \mathbf{f}^{(k)}$ and for $j=1,\dots,p$ compute:

$$\mathbf{h}^{(j)} = \mathbf{h}^{(j-1)} \frac{U_{0j}(\mathbf{h}^{(j-1)}; \mathbf{g}_j) + \mu/p U_1(\mathbf{h}^{(j-1)})}{V_{0j}(\mathbf{h}^{(j-1)}; \mathbf{g}_j) + \mu/p V_1(\mathbf{h}^{(j-1)})}$$

- set $\mathbf{f}^{(k+1)} = \mathbf{h}^{(p)}$.

In the following table we give the expression of U_{0j} and V_{0j} :

noise	U_{0j}	V_{0j}
(P)	$\mathbf{A}_j^T \frac{\mathbf{g}_j}{\mathbf{A}_j \mathbf{h}^{(j)} + \mathbf{b}_j}$	$\boldsymbol{\alpha}_j$
(G)	$\mathbf{A}_j^T \mathbf{g}_j$	$\mathbf{A}_j^T \mathbf{A}_j \mathbf{h}^{(j)} + \mathbf{b}_j$

Table 3. OS-versions of the algorithms implemented in AIRY. The definition of the arrays $\boldsymbol{\alpha}_j$ is given in Eq. 8.

Remarks.

1. The algorithm can be initialized by an estimate of the unknown object, if any, or a constant array $\mathbf{f}^{(0)}$ that, for convenience, is chosen satisfying the so-called *flux condition*:

$$\frac{1}{p} \sum_{\mathbf{n} \in R} \boldsymbol{\alpha} \mathbf{f}^{(0)}(\mathbf{n}) = \frac{1}{p} \sum_{j=1}^p \sum_{\mathbf{m} \in S} (\mathbf{g}_j - \mathbf{b}_j)(\mathbf{m}) = c \quad .$$

2. Concerning the reconstruction region R , it depends on the problem and, in particular, on the PSF, and is defined as follows:

$$\text{RL and OSEM:} \quad R = \{\mathbf{n} | \boldsymbol{\alpha}(\mathbf{n}) > \sigma \cdot p\} \quad , \quad (9)$$

$$\text{ISRA and OS-ISRA:} \quad R = \{\mathbf{n} | \sum_{j=1}^p \mathbf{A}_j^T \mathbf{g}_j(\mathbf{n}) > \sigma \cdot \bar{c}\} \quad , \quad (10)$$

where $\sigma < 1$ is a given thresholding parameter and \bar{c} is the average flux per pixel (we denote with $\#(S)$ the number of pixel in S) defined by

$$\bar{c} = \frac{1}{p \cdot \#(S)} \sum_{j=1}^p \sum_{\mathbf{n} \in S} \mathbf{g}_j \quad .$$

2.3 Accelerations

The previous algorithms can be easily accelerated following the scheme proposed by Biggs and Andrews^{37,38}, which can be applied to algorithms of the form $\mathbf{f}^{(k+1)} = T(\mathbf{f}^{(k)})$, with a linear or nonlinear operator T independent of k . In the **Software Package** AIRY we implemented two different versions of accelerations, namely the *linear* and the *exponential* acceleration. The basic idea is to attempt an extrapolation of the current iterate, using the result of the current iteration and of the previous one. For more details see La Camera et al.²². Here we report the exponential version of the algorithm, where the operator T is one of the standard algorithms described above.

- Initialize the algorithm with $\mathbf{f}^{(0)} > 0$ as discussed in the previous remark.
- Compute and store: $\mathbf{e}^{(0)} = \mathbf{f}^{(0)}$, $\mathbf{e}^{(1)} = \mathbf{f}^{(1)} = T(\mathbf{f}^{(0)})$ and $\mathbf{f}^{(2)} = T(\mathbf{f}^{(1)})$.
- For $k=2,3,\dots$ given the four arrays $\{\mathbf{f}^{(k)}, \mathbf{f}^{(k-1)}, \mathbf{e}^{(k-1)}, \mathbf{e}^{(k-2)}\}$ compute:

$$\Delta^{(k-1)} = \frac{\mathbf{f}^{(k-1)}}{\mathbf{e}^{(k-2)}}, \quad \Delta^{(k)} = \frac{\mathbf{f}^{(k)}}{\mathbf{e}^{(k-1)}}, \quad \lambda_k = \frac{(\mathbf{f}^{(k)} \log \Delta^{(k)}, \mathbf{f}^{(k-1)} \log \Delta^{(k-1)})}{\|\mathbf{f}^{(k-1)} \log \Delta^{(k-1)}\|^2}, \quad (0 \leq \lambda_k \leq 1),$$

$$\mathbf{e}^{(k)} = \mathbf{f}^{(k)} \left(\frac{\mathbf{f}^{(k)}}{\mathbf{f}^{(k-1)}} \right)^{\lambda_k}.$$

- Compute $\mathbf{f}^{(k+1)} = T(\mathbf{e}^{(k)})$.

Work is in progress for implementing the *Scaled Gradient Projection* (SGP) algorithm proposed by Bonettini et al.³⁹.

2.4 Stopping criteria

Except in the case of point-wise objects, early stopping of the iterations is necessary for obtaining sensible solutions in the absence of a regularization functional ($\mu = 0$). Thus, stopping rules must be used. In the **Software Package AIRY** we implemented four stopping criteria for both RL and OSEM algorithms:

1. Set a total number of iterations and stop the algorithm when this number is reached. This is valid for all available methods;
2. Stop the iteration when the discrepancy function

$$D^{(k)} = \frac{2}{p \cdot \#(S)} J_0(\mathbf{f}^{(k)}; \mathbf{g})$$

crosses 1. We denote by $\#(S)$ the number of pixels in S . This criterion is called *discrepancy principle* for Poisson data⁴⁰ and can be used when $\mu = 0$ (i.e. no regularization has been used).

3. Stop the iterations when the total functional $J_\mu(\mathbf{f}^{(k)}; \mathbf{g})$ is approximately constant, according to a user-defined tolerance. This stopping rule can be applied only when a regularization is chosen ($\mu \neq 0$).
4. Stop the iterations when the relative r.m.s error

$$\rho^{(k)} = \frac{\|\mathbf{f}^{(k)} - \bar{\mathbf{f}}\|}{\|\bar{\mathbf{f}}\|}$$

reaches a minimum value. This stopping rule can be used only in the case of numerical simulations, when the true object $\bar{\mathbf{f}}$ is known.

The first and the last stopping rules are also available for ISRA and OS-ISRA.

2.5 Super-resolution

As known, RL and OSEM have a modest super-resolution effect. However it is possible to increase this effect by a suitable initialization of the algorithms. Indeed, if the object has an angular size comparable with the angular resolution of the instrument, it is possible to improve the resolution by considering the reconstruction within this domain. This can be obtained by masking the initial guess with a function that is 1 in the super-resolution domain and zero elsewhere. The pixels with zero values will not change in all iterations. A discussion of the method can be found in Anconelli et al.¹³. The super-resolution method is explained in two example projects provided with the software package and described in Sect. 4. The module **DEC** of **AIRY** permits to perform reconstructions with super-resolution.

2.6 Blind Deconvolution

As described in Desiderà et al.^{17,18}, the iterative blind deconvolution (IBD) method restores the object and the PSF separately in an iterative form. Within each “global” iteration, either the object (within the so-called *object box*) or the PSF (in the *PSF box*) is reconstructed while the other one is kept fixed. In other words, within the object box, the object is reconstructed by using the updated PSF and vice-versa in the PSF box. Therefore the output of the global iteration updates both the object and the PSF.

In the two boxes we use the RL algorithm to perform the reconstruction with a number of iterations that is in general different for the object and the PSF reconstructions. We denote again by k the global iterations and by i_{obj} and i_{psf} the number of iterations within each box. In general $k \neq i_{obj} \neq i_{psf}$. In particular, a bad choice for i_{obj} or i_{psf} can compromise the result of the deconvolution for both the object and the PSF. For example, performing too many iterations on the PSF can over-fit the peak of the PSF with respect to the “true” PSF. On the other hand, choosing a too small number of iterations could be insufficient to improve the reconstructed PSF.

In order to prevent the risk of choosing too many iterations for i_{psf} we introduced a Strehl⁴¹ constraint on the reconstruction of the PSF. It consists in blurring the output of the PSF box when the Strehl ratio (SR) of the current PSF exceeds the pre-fixed SR. The blurring is performed by an iterative procedure consisting in convolving the PSF with a Gaussian function with a small FWHM, in order to reduce the peak of the PSF to a desired value. The Gaussian function is normalized to the unit volume and this preserves the normalization of the PSF. The entire process is described in Desiderà et al.¹⁸ and it is implemented in the module `CBD` of `AIRY`.

2.7 PSF extraction and extrapolation

As it is known, the PSF is not always known in the restoration problem, especially in the case of ground-based telescopes equipped with adaptive optics systems. In such a case, the PSF can be extracted from the input image if a star is present in the Field of View of the instrument. The image of the star has a smaller size than the full image and it could be extended by zero-padding, but this extension may not be suitable for image deconvolution. For this reason we implemented an extrapolation of the PSF outside the extraction domain by means of a suitable Moffat function with the form:

$$M(r) = \frac{a}{(b + r^2)^\beta} \quad ,$$

where r is the distance from the centre (the star), a and b are parameters to be fitted and β is the exponent of the Moffat function. The procedure is described in La Camera et al.²² and is implemented within the module `PEX` of `AIRY`.

3. THE SOFTWARE PACKAGE AIRY

3.1 Description of the Software Package

As mentioned in Sect. 1, `AIRY` has a modular structure, i.e. it consists of *modules*, each one performing a specific task, and is designed to be used within the `CAOS PSE` (also called “`CAOS system`”). `CAOS` (short for Code for Adaptive Optics Systems) provides the global interface and the global structure of the tool, permitting to complete the entire tool with a number of other packages covering a wider area of astronomical-optics-related scientific topics. A description of the available software packages for the `CAOS PSE` can be found in Carillet et al.^{8,9}. The `CAOS PSE` is composed of a global graphical interface, the `CAOS Application Builder`, which permits to connect together modules from the various installed packages in order to define a simulation data flow. The `CAOS PSE` also contains a library of utilities, namely the `CAOS Library`, and, of course, the `Software Packages`, each of them being a collection of modules. There are no limits to the installable packages, so the `CAOS PSE` may be always updated with new packages.

In Tab. 4 we give a brief description of the modules included in the `Software Package AIRY`, divided in three categories: simulation modules (for data simulation), deconvolution and analysis modules (for performing data processing and analyzing the results), and other modules (for input/output operations and other utilities).

	Module	Purpose
Simulation	OBJ - OBJECT definition	To define the object in a simulation.
	CNV - object-psf CoNVolution	To perform the convolution between the object and the PSF. To set the acquisition time, the number of frames, the telescope surface and the overall efficiency.
	ADN - Add Noise to Image	To set the background value; to add the Poisson and Gaussian noise; to simulate flat-field distortion and bad-pixels.
Deconvolution and data analysis	PRE - PRE-processing	To perform all pre-processing steps before reconstructing the images; in particular the background estimation must be set up in this module.
	PEX - Psf EXtraction	To extract and extrapolate the PSF(s) from a star contained in the input image(s).
	DEC - DEConvolution process	To perform the deconvolution of the images by using different algorithms, boundary effects correction, regularizations, accelerations, and stopping rules.
	CBD - Constrained Blind Deconvolution	To reconstruct both the target and the PSF, by using a priori information about acquisition band and Strehl ratio.
	ANB - ANalysis Binary	To analyse a reconstructed image of a binary star.
	FSM - Find Star Module	To analyse a reconstructed image and retrieve information about the stars contained on the field of view.
I/O modules and utilities	RFT - Read FiTs file	To read images (or data) from a FITS file.
	RSC - Restore image Structure Cubes	To restore data structures saved with SIM module.
	DSP - image DiSPlay	To show the input image.
	RTI - RoTate Image	To rotate objects or images in two different ways: in the pixel space (by using the standard routine ROT) and in the Fourier space.
	SIM - Save IMage structure	To save and export data structure.
	WFT - Write FiTs file	To save images (or data) into a FITS file.

Table 4. The descriptive list of AIRY modules for the upcoming version 6.0.

3.2 What is new in AIRY

In the forthcoming version of AIRY, several new features have been included. The first, and the most important one, is the implementation of all reconstruction algorithms described in the previous sections.

The module DEC - (Deconvolution process) is now able to deconvolve the input image by using 4 different algorithms (RL, OSEM, ISRA, and OS-ISRA) and each one can be executed with or without the correction of boundary effects. Moreover, each algorithm can be regularized (choosing among 5 different regularization functionals) and accelerated in two different ways. The algorithms have been re-written, with respect to the previous version, keeping in mind also the performance: the new module is faster than the previous one (see next section for details). In Fig. 1 we show the new graphical user interface (GUI) of the module, where the algorithm and all the necessary parameters can be set up. In Fig. 2 we show the second panel of the GUI where the stopping rules take place.

In the new version of AIRY we improved the module PEX - (Psf EXtraction). The module is used to extract and extrapolate the PSF from one star in the input image. As described in Sect. 2.7, the extrapolation is performed by fitting the extracted data with a Moffat function. The value of β must be provided by the user and depends on the sky condition during the acquisition. The new feature of PEX permits to compute the value of β by an automatic procedure, assuming that the acquisition parameters are known (magnitude of the star, time integration, surface of the telescope, background, etc.). In Fig. 3 we show the GUI of the module. More

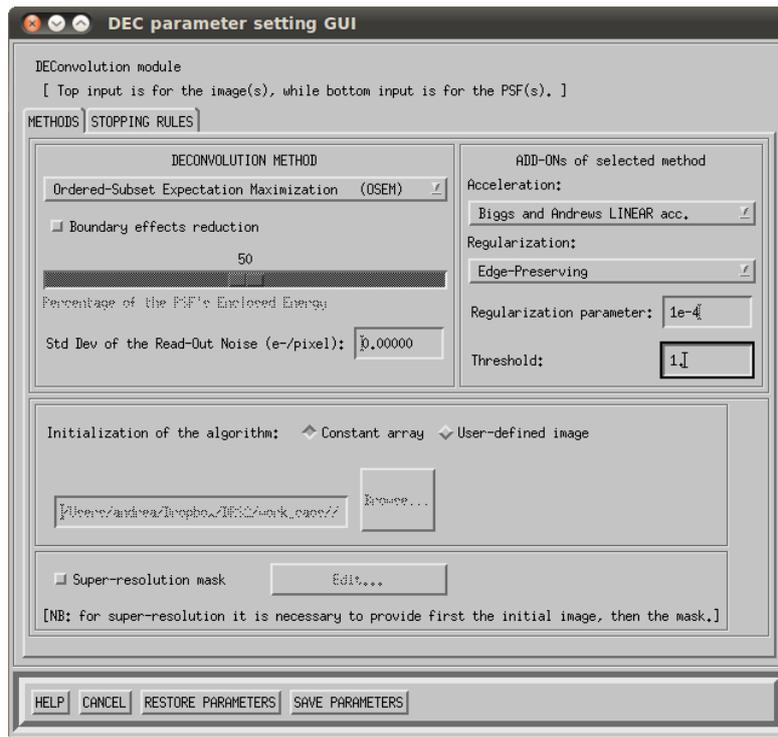


Figure 1. The graphical user interface (GUI) of the module DEC, first panel (algorithms and other parameters).

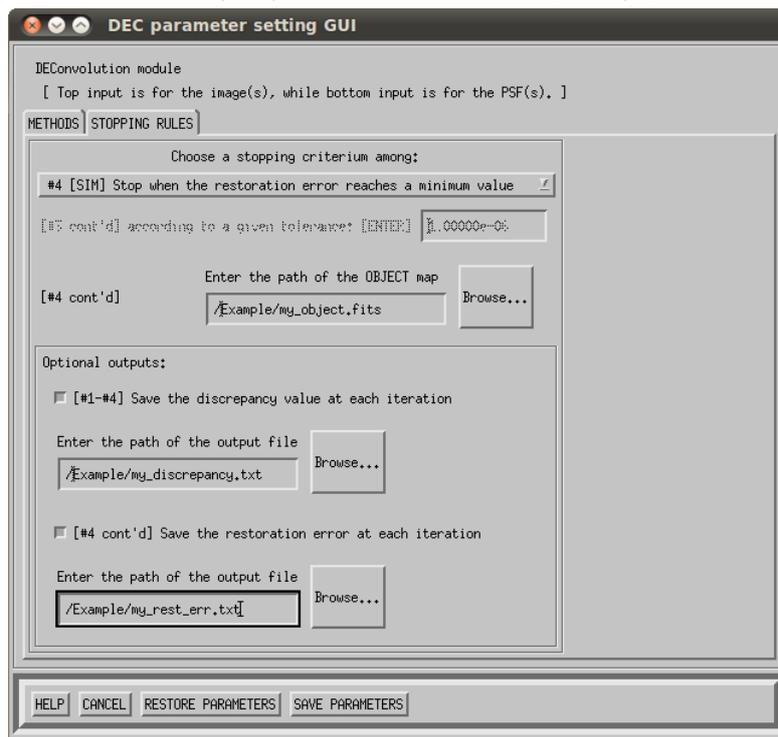


Figure 2. The graphical user interface (GUI) of the module DEC, second panel (stopping rules).

information on this automatic procedure and a discussion of the importance of the β parameter can be found in Ciliegi et al.⁴².

Several improvements, bugs corrections and general checks have been performed in each module, in order to complete this new major release of our package.

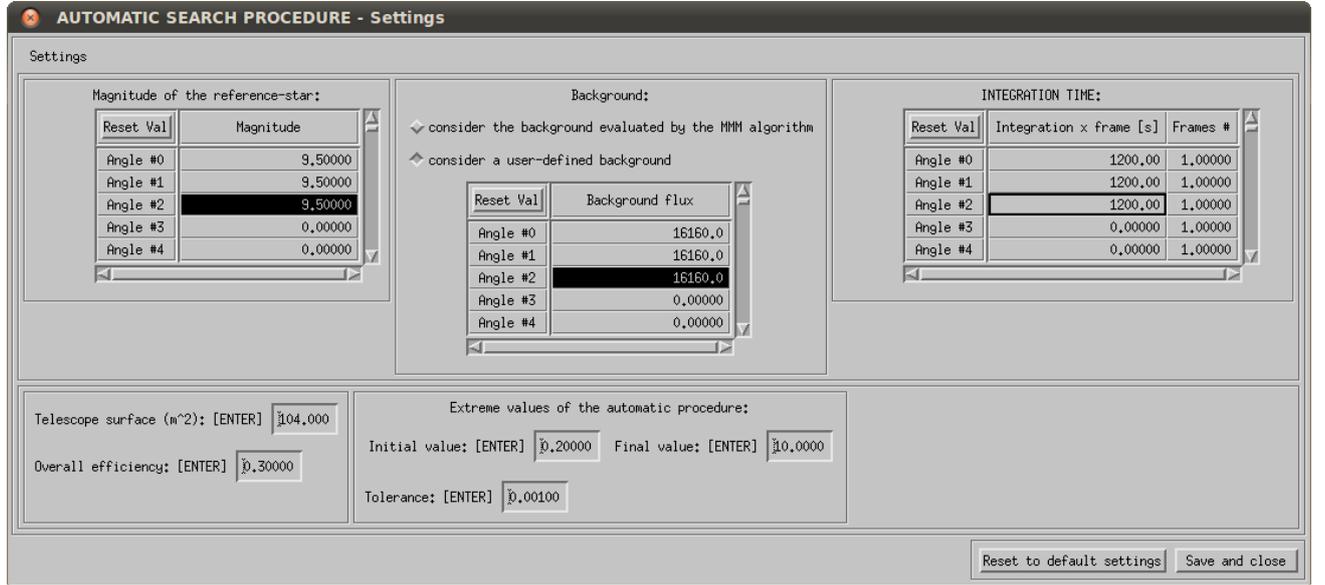


Figure 3. The sub-GUI of the module PEX, where requested parameters are set up for the automatic searching procedure.

4. EXAMPLE PROJECTS

The **Software Package** AIRY is provided with 8 different example projects, described in Tab. 5. Six of them were already available in the previous version of AIRY, one is a modification of an existent project (namely “PSF_Extraction_Example”), while the project called “StellarSurface_Example” is a new one. For each project using the deconvolution module (DEC), we measured the execution time for both the old and the new version of AIRY on the same PC and we provided the speed-up factor in the last column of the table. As mentioned in Sect. 3, within the module DEC we totally re-wrote the code of the algorithms, paying attention to efficiency.

The “PSF_Extraction_Example” project contains a simulation of a star cluster, i.e. the convolution of 15 point-like sources, with magnitudes between 10 and 15 in the K band, with an AO PSF. A list of star positions and magnitudes is provided together with the project. From the image of the star cluster, by using the module PEX, we extracted and extrapolated the PSF by using the new automatic searching procedure for the β parameter.

The “StellarSurface_Example” project concerns the reconstruction of a stellar surface image by using two different algorithms: OSEM without regularization and OSEM with Edge-Preserving regularization. Both algorithms are accelerated by means of Biggs and Andrews exponential acceleration. The two parallel reconstructions show how important is, in such a case, to use a regularized version of the algorithm: the comparison of the two restored objects with the true object used for simulating the image is shown in Fig. 4.

5. CONCLUSIONS AND FUTURE DEVELOPMENTS

We presented here the **Software Package** AIRY, a freely downloadable tool for simulation and deconvolution of astronomical images. In this paper we described the reconstruction algorithms available in AIRY as well as the structure of the package and the main features of the new version of the package. Moreover we described 8 different example projects that are also downloadable together with AIRY.

In conclusion, we briefly discuss the future of the **Software Package** AIRY in a few points:

Project Name	Purpose	Speed-up
1. Binary_Data_Example	To deconvolve a simulated image of a binary star system by using a given PSF. The reconstructed object, after 250 iterations, is saved on “results” folder, together with the analysis of the binary performed by the module ANB.	1.22x
2. High_Dynamic_Object_Example	To simulate and deconvolve a high dynamic range object, consisting of a binary star surrounded by a diffuse ring. The reconstruction uses the HDR regularized version of OSEM.	2.57x
3. Rotate_Image_Example	To rotate the input image by using two different procedures: the rotation in the pixel space (faster) and the rotation in the Fourier domain (more precise, but if used with point-like objects can generate artifacts).	–
4. StrehlConstrained_Example	To simulate and deconvolve a binary star image. The simulated image is used to reconstruct both the object and the PSF by using the module CBD. Two different CBDs are used within this project, one without constraints and the other one with Strehl constraint on the reconstructed PSF.	–
5. SuperResolution_Step1	To simulate and deconvolve a binary star by using the super-resolution technique. The project is subdivided in two...	1.16x
6. SuperResolution_Step2	...consecutive steps: the reconstructed object obtained at the end of step1 is used as an estimate for initializing the algorithm of step2.	1.17x
7. PSF_Extraction_Example	Description in the text.	–
8. StellarSurface_Example	Description in the text.	–

Table 5. The eight example projects provided together with the software package AIRY.

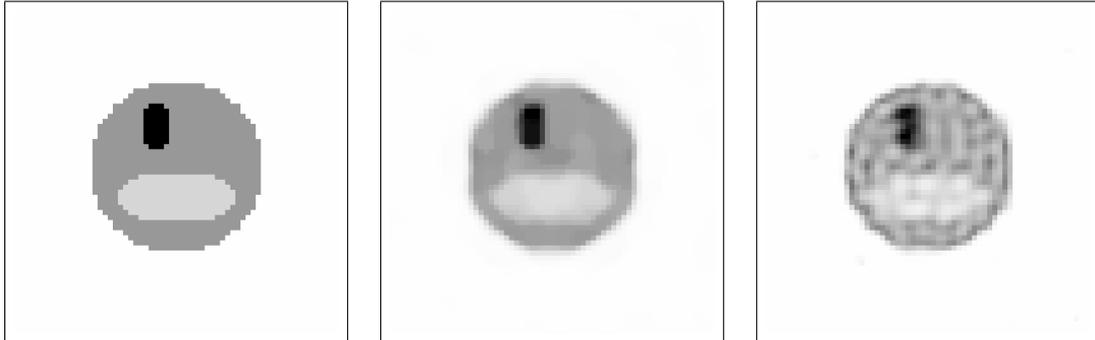


Figure 4. The true object (left), the restored object with edge-preserving regularization (middle) and the restored object without regularization (right).

- From its beginning, AIRY has been developed for the reconstruction of astronomical images. After more than ten years of advances, the deconvolution module DEC has four different reconstruction algorithms, boundary effects correction, five regularizations, two accelerations and four stopping rules. Work is in progress for implementing the SGP³⁹ algorithm side by side with the existent algorithms.
- In the last years, we also focused on the PSF reconstruction (or estimation). For this reason we developed techniques for extraction and extrapolation of the PSF and, on the other hand, for blind deconvolution in which we can reconstruct both the PSF and the object. Further developments in this sense will also consider a denoising step before extracting the PSF and using it in the deconvolution step. Indeed, we already successfully experimented⁷ the use of BLS-GSM (Bayesian Least Square Gaussian Scale Mixture) for PSF restoration. BLS-GSM is a wavelet-based denoising method and it was first presented by Portilla et al.⁴³. BLS-SGM is coupled with the Full Steerable Pyramid⁴⁴, to decompose the input image in a set

of wavelet coefficients with an high orientation adaptivity. The algorithm is finally adapted to deal with Poisson noise, introducing a variance-stabilizing transform, the Anscombe transform, to approximately convert the Poisson noise to Gaussian noise with unitary variance, and its unbiased inverse Anscombe transform recently presented in Makitalo et al.⁴⁵. Our preliminary results⁷ show that the accuracy of the reconstruction obtained by using a PSF extracted from a very faint star and “cleaned” with BLS-GSM is comparable with the one provided by an exact PSF.

6. AVAILABILITY OF THE CODE

The Software Package AIRY as well as the whole CAOS PSE can be freely downloaded from:

<http://lagrange.oca.eu/caos/>

ACKNOWLEDGMENTS

This work has been partially supported by MIUR (Italian Ministry for University and Research), PRIN2008 “Optimization Methods and Software for Inverse Problems”, grant 2008T5KA4L, and by INAF (National Institute for Astrophysics) under the contract TECNO-INAf 2010 “Exploiting the adaptive power: a dedicated free software to optimize and maximize the scientific output of images from present and future adaptive optics facilities”.

REFERENCES

- [1] Correia, S., Carbillet, M., Fini, L., Boccacci, P., Bertero, M., Vallenari, A., Richichi, A., and Barbati, M., “AIRY: Astronomical Image Restoration in interferometrY,” in [*Astronomical Data Analysis Software and Systems X*], Harnden, Jr., F. R., Primini, F. A., and Payne, H. E., eds., *Astronomical Society of the Pacific Conference Series* **238**, 404 (2001).
- [2] Correia, S., Carbillet, M., Boccacci, P., Bertero, M., and Fini, L., “Restoration of interferometric images. I. The software package AIRY,” *Astronomy and Astrophysics* **387**, 733–743 (May 2002).
- [3] Bertero, M. and Boccacci, P., “Application of the OS-EM method to the restoration of LBT images,” *Astronomy and Astrophysics Supplement* **144**, 181–186 (May 2000).
- [4] Bertero, M. and Boccacci, P., “Image restoration methods for the Large Binocular Telescope (LBT),” *Astronomy and Astrophysics Supplement* **147**, 323–333 (Dec. 2000).
- [5] Correia, S., Carbillet, M., Richichi, A., Bertero, M., and Boccacci, P., “Large Binocular Telescope image restoration using simulated adaptively corrected point-spread functions,” in [*Interferometry in Optical Astronomy*], Léna, P. and Quirrenbach, A., eds., *Proc. SPIE* **4006**, 650–658 (July 2000).
- [6] Bertero, M., Boccacci, P., Desiderà, G., La Camera, A., Carbillet, M., and Lantéri, H., “Imaging with LINC-NIRVANA,” *IEEE Signal Processing Magazine* **27**, 110–115 (Jan. 2010).
- [7] Bertero, M., Boccacci, P., La Camera, A., Olivieri, C., and Carbillet, M., “Imaging with LINC-NIRVANA, the Fizeau interferometer of the Large Binocular Telescope: state of the art and open problems,” *Inverse Problems* **27**(11), 113001 (2011).
- [8] Carbillet, M., Verinaud, C., Guarracino, M., Fini, L., Lardiere, O., Le Roux, B., Puglisi, A. T., Femenia, B., Riccardi, A., Anconelli, B., Correia, S., Bertero, M., and Boccacci, P., “CAOS: a numerical simulation tool for astronomical adaptive optics (and beyond),” in [*Advancements in Adaptive Optics*], Bonaccini Calia, D., Ellerbroek, B. L., and Ragazzoni, R., eds., *Proc. SPIE* **5490**, 637–648 (Oct. 2004).
- [9] Carbillet, M., Desiderà, G., Augier, E., La Camera, A., Riccardi, A., Boccaletti, A., Jolissaint, L., and Ab Kabir, D., “The CAOS problem-solving environment: recent developments,” in [*Adaptive Optics Systems II*], Ellerbroek, B. L., Hart, M., Hubin, N., and Wizinowich, P. L., eds., *Proc. SPIE* **7736** (July 2010).
- [10] Carbillet, M., Verinaud, C., Femenia, B., Riccardi, A., and Fini, L., “Modelling astronomical adaptive optics - I. The software package CAOS,” *Monthly Notices of the Royal Astronomical Society* **356**, 1263–1275 (Feb. 2005).
- [11] Carbillet, M., Correia, S., Boccacci, P., and Bertero, M., “Restoration of interferometric images. II. The case-study of the Large Binocular Telescope,” *Astronomy and Astrophysics* **387**, 744–757 (May 2002).

- [12] Anconelli, B., Bertero, M., Boccacci, P., Carbillet, M., and Lantéri, H., “Restoration of interferometric images. III. Efficient Richardson-Lucy methods for LINC-NIRVANA data reduction,” *Astronomy and Astrophysics* **430**, 731–738 (Feb. 2005).
- [13] Anconelli, B., Bertero, M., Boccacci, P., and Carbillet, M., “Restoration of interferometric images. IV. An algorithm for super-resolution of stellar systems,” *Astronomy and Astrophysics* **431**, 747–755 (Feb. 2005).
- [14] Anconelli, B., Bertero, M., Boccacci, P., Carbillet, M., and Lantéri, H., “Reduction of boundary effects in multiple image deconvolution with an application to LBT LINC-NIRVANA,” *Astronomy and Astrophysics* **448**, 1217–1224 (Mar. 2006).
- [15] Anconelli, B., Bertero, M., Boccacci, P., Desiderà, G., Carbillet, M., and Lantéri, H., “Deconvolution of multiple images with high dynamic range and an application to LBT LINC-NIRVANA,” *Astronomy and Astrophysics* **460**, 349–355 (Dec. 2006).
- [16] Anconelli, B., Bertero, M., Boccacci, P., Carbillet, M., and Lantéri, H., “Iterative methods for the reconstruction of astronomical images with high dynamic range,” *Journal of Computational and Applied Mathematics* **198**, 321–331 (Jan. 2007).
- [17] Desiderà, G., Anconelli, B., Bertero, M., Boccacci, P., and Carbillet, M., “Application of iterative blind deconvolution to the reconstruction of LBT LINC-NIRVANA images,” *Astronomy and Astrophysics* **452**, 727–734 (June 2006).
- [18] Desiderà, G. and Carbillet, M., “Strehl-constrained iterative blind deconvolution for post-adaptive-optics data,” *Astronomy and Astrophysics* **507**, 1759–1762 (Dec. 2009).
- [19] Habart, E., Testi, L., Natta, A., and Carbillet, M., “Diamonds in HD 97048: A Closer Look,” *The Astrophysical Journal* **614**, L129–L132 (Oct. 2004).
- [20] Habart, E., Natta, A., Testi, L., and Carbillet, M., “Spatially resolved PAH emission in the inner disks of Herbig Ae/Be stars,” *Astronomy and Astrophysics* **449**, 1067–1075 (Apr. 2006).
- [21] Domiciano de Souza, A., Kervella, P., Bendjoya, P., and Niccolini, G., “High-resolution thermal infrared imaging of MWC300. VLT/VISIR observations in BURST mode,” *Astronomy and Astrophysics* **480**, L29–L32 (Mar. 2008).
- [22] La Camera, A., Desiderà, G., Arcidiacono, C., Boccacci, P., and Bertero, M., “Advances in the reconstruction of LBT LINC-NIRVANA images,” *Astronomy and Astrophysics* **471**, 1091–1097 (2007).
- [23] Desiderà, G., La Camera, A., Boccacci, P., Bertero, M., and Carbillet, M., “AIRY-LN: an ad-hoc numerical tool for deconvolution of images from the LBT instrument LINC-NIRVANA,” in [*Optical and Infrared Interferometry*], Schoeller, M., Danchi, W. C., and Delplancke, F., eds., *Proc. SPIE* **7013** (July 2008).
- [24] Anconelli, B., Bertero, M., Boccacci, P., Carbillet, M., Lantéri, H., and Correia, S., “Deconvolution methods for LINC/NIRVANA data reduction,” in [*New Frontiers in Stellar Interferometry*], Traub, W. A., ed., *Proc. SPIE* **5491**, 932 (Oct. 2004).
- [25] Snyder, D. L., Hammoud, A. M., and White, R. L., “Image recovery from data acquired with a charge-coupled-device camera,” *Journal of the Optical Society of America A* **10**, 1014–1023 (May 1993).
- [26] Benvenuto, F., La Camera, A., Theys, C., Ferrari, A., Lantéri, H., and Bertero, M., “The study of an iterative method for the reconstruction of images corrupted by Poisson and Gaussian noise,” *Inverse Problems* **24**(3), 035016, 20 (2008).
- [27] Snyder, D. L., Helstrom, C. W., Lanterman, A. D., Faisal, M., and White, R. L., “Compensation for readout noise in CCD images,” *Journal of the Optical Society of America A: Optics* **12**, 272–283 (Feb. 1995).
- [28] Thiebaut, E. and Giovannelli, J.-F., “Image reconstruction in optical interferometry,” *Signal Processing Magazine, IEEE* **27**(1), 97–109 (2010).
- [29] Csiszár, I., “Why least squares and maximum entropy? An axiomatic approach to inference for linear inverse problems,” *The Annals of Statistics* **19**(4), 2032–2066 (1991).
- [30] Lantéri, H., Roche, M., and Aime, C., “Penalized maximum likelihood image restoration with positivity constraints: multiplicative algorithms,” *Inverse Problems* **18**, 1397 (2002).
- [31] Richardson, W. H., “Bayesian-Based Iterative Method of Image Restoration,” *Journal of the Optical Society of America* **62**, 55 (Jan. 1972).
- [32] Lucy, L. B., “An iterative technique for the rectification of observed distributions,” *Astronomical Journal* **79**, 745 (June 1974).

- [33] Daube-Witherspoon, M. E. and Muehlehner, G., “An Iterative Image Space Reconstruction Algorithm Suitable for Volume ECT,” *Medical Imaging, IEEE Transactions on* **5**(2), 61–66 (1986).
- [34] Bertero, M. and Boccacci, P., “A simple method for the reduction of boundary effects in the Richardson-Lucy approach to image deconvolution,” *Astronomy and Astrophysics* **437**, 369 (July 2005).
- [35] Shepp, L. A. and Vardi, Y., “Maximum Likelihood Reconstruction for Emission Tomography,” *Medical Imaging, IEEE Transactions on* **1**(2), 113–122 (1982).
- [36] Hudson, H. M. and Larkin, R. S., “Accelerated image reconstruction using ordered subsets of projection data,” *Medical Imaging, IEEE Transactions on* **13**(4), 601–609 (1994).
- [37] Biggs, D. S. C. and Andrews, M., “Acceleration of iterative image restoration algorithms,” *Applied Optics* **36**, 1766 (Mar. 1997).
- [38] Biggs, D. S. and Andrews, M., “Asymmetric iterative blind deconvolution of multiframe images,” in [*Advanced Signal Processing Algorithms, Architectures, and Implementations VIII*], Luk, F. T., ed., *Proc. SPIE* **3461**, 328–338 (Oct. 1998).
- [39] Bonettini, S., Zanella, R., and Zanni, L., “A scaled gradient projection method for constrained image deblurring,” *Inverse Problems* **25**, 015002 (2009).
- [40] Bertero, M., Boccacci, P., Talenti, G., Zanella, R., and Zanni, L., “A discrepancy principle for Poisson data,” *Inverse Problems* **26**, 5004 (Oct. 2010).
- [41] Strehl, K., “Über luftschlieren und zonenfehler,” *Zeitschrift für Instrumentenkunde* **22**, 213–217 (1902).
- [42] Ciliegi, P., La Camera, A., Arcidiacono, C., Bertero, M., Boccacci, P., Diolaiti, E., Foppiani, I., Lombini, M., and Schreiber, L., “Analysis of LBT LINC-NIRVANA simulated images of galaxies,” in [*Optical and Infrared Interferometry II*], Danchi, W. C., Delplancke, F., and Rajagopal, J. K., eds., *Proc. SPIE* **7734** (July 2010).
- [43] Portilla, J., Strela, V., Wainwright, M., and Simoncelli, E., “Image denoising using scale mixtures of gaussians in the wavelet domain,” *Image Processing, IEEE Transactions on* **12**, 1338–1351 (Nov. 2003).
- [44] Simoncelli, E. and Freeman, W., “The steerable pyramid: a flexible architecture for multi-scale derivative computation,” in [*Image Processing, 1995. Proceedings., International Conference on*], **3**, 444–447 (oct 1995).
- [45] Makitalo, M. and Foi, A., “Optimal inversion of the Anscombe transformation in low-count Poisson image denoising,” *Image Processing, IEEE Transactions on* **20**(1), 99–109 (2010).