Deconvolution methods for LINC/NIRVANA data reduction

Anconelli B.[a], Bertero M.[a], Boccacci P.[a], Carbillet M.[b], Lanteri H.[c] and Correia S.[d]

[a]INFM and DISI, Università di Genova, Via Dodecaneso 35, 16146 Genova, Italy;
[b]INAF-Osservatorio Astrofisico di Arcetri, Largo E. Fermi 5, 50125 Firenze, Italy;
[c]UMR 6525 Astrophysique, Université de Nice - Sophia Antipolis, Faculté des Sciences - Parc Valrose, 06108 Nice Cedex 02, France;
[d]Astrophysikalisches Institut Potsdam, An der Sternwarte 16, 14482 Potsdam, Germany

ABSTRACT

LINC/NIRVANA (LN) is the German-Italian beam combiner for the Large Binocular Telescope (LBT). It is a Fizeau interferometer and it will provide multiple images of the same target corresponding to different orientations of the baseline. For each one of these images the resolution is not uniform over the field since it is the resolution of a $22.8m$ mirror in the direction of the baseline and that of a $8.4m$ mirror in the orthogonal one. Therefore a unique high-resolution image can only be obtained by means of deconvolution methods. Four-years ongoing work of our group on this problem has already clarified the effects of partial adaptive optics corrections and partial coverage of the u, v plane and has produced the Software Package AIRY, a set of modules IDL-based and CAOS-compatible, which can be used for simulation and/or deconvolution of multiple images from the LBT instrument LN. In this paper we present a general approach to the design of methods for the simultaneous deconvolution of multiple images of the same object. These can include both quick-look methods, to be used for routinely process LN images, and ad-hoc methods for specific classes of astronomical objects. We describe several examples of these methods whose implementation and validation is in progress. Finally we present the last version of the Software Package AIRY.

Keywords: Interferometric imaging - Data reduction - Large Binocular Telescope - LINC/NIRVANA

1. INTRODUCTION

In previous papers (Correia et al.,[11] Carbillet et al.,[8] Carbillet et al.,[9] Anconelli et al.[1]) we developed methods and software for the application of image deconvolution to Fizeau interferometers. An example will be provided by LINC/NIRVANA (LN), the German-Italian beam combiner for the Large Binocular Telescope (LBT). LBT will consist of two $8.4m$ mirrors on a common mount, with a spacing of $14.4m$ between the centers of the two mirrors, so that a maximum baseline of $22.8m$ will be available. First light of LBT is scheduled for September 2004, second light for November 2005, and the first LN light is foreseen for July 2006.

The interferometric technique used in LN will provide direct imaging with the resolution of a $22.8m$ telescope in the direction of the baseline and of a $8.4m$ telescope in the orthogonal direction. Since resolution is not uniform over the field, several images of the same scientific object must be acquired with different orientations of the baseline and they must be suitably processed in order to get a unique image with a uniform resolution over the field. Therefore imaging by LN will require routinely the use of multiple-images deconvolution methods if the goal is a unique image with the resolution of a $22.8m$ telescope. Our group has already produced the Software Package AIRY (Astronomical Image Restoration in interferometrY) which is now available in the version 2.1 (see `http://dirac.disi.unige.it` and `http://www.arcetri.astro.it/caos`). AIRY is a set of modules for the simulation and/or analysis of a set of interferometric images, including a deconvolution module implementing the OSEM (Ordered Subset Expectation Maximization) method, which is an accelerated version of the extension of the RL (Richardson-Lucy) method to multiple-images deconvolution.[5] In this paper we present 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.

Send correspondence to Bertero M. E-mail: bertero@disi.unige.it

In the following we use bold letters for $N \times N$ arrays. Therefore we denote by $\mathbf{g}_1, \mathbf{g}_2, ..., \mathbf{g}_p$ the p LN images of a given astronomical object, corresponding to p different orientations of the LBT baseline; moreover we often use the more compact notation $\mathbf{g}$ for indicating the set of the p images. Each one of these images, acquired with a CCD camera, can be described by the following model (Snyder et al.[20]):

$$\mathbf{g}_j(m,n) \;=\; \mathbf{g}_{obj,j}(m,n) \;+\; \mathbf{g}_{back,j}(m,n) \;+\; \mathbf{r}_j(m,n) \quad , \quad j = 1, 2, ..., p \quad , \tag{1}$$

where $\mathbf{g}_j(m,n)$ denotes the value of the detected j-th image at pixel m, n; it is given by the number of photoelectrons due to object radiation, $\mathbf{g}_{obj,j}(m,n)$, and to background radiation, $\mathbf{g}_{back,j}(m,n)$, as well as by the read-out noise due to the amplifier, $\mathbf{r}_j(m,n)$. The first two terms are realizations of independent Poisson processes, so that their sum is also a Poisson process while the third term is the realization of a Gaussian process (white noise). The expected value of the j-th image is given by:

$$E\{\mathbf{g}_j\} \;=\; A_j\mathbf{f} \;+\; \mathbf{b}_j \quad , \tag{2}$$

where: $\mathbf{f}$ is the unknown object, an array formed by the average numbers of photons emitted at the pixels of the object domain and collected by the telescope; A_j is the matrix describing the imaging process (optics plus atmosphere); $\mathbf{b}_j$ is a background term, which can include a non-zero expected value of the read-out noise and, in general, can be assumed to be constant over the image domain. If we assume, for simplicity, space invariance, the matrix A_j can be given in terms of a point spread function (PSF) $\mathbf{K}_j$:

$$A_j\mathbf{f} \;=\; \mathbf{K}_j * \mathbf{f} \quad ; \tag{3}$$

moreover we suppose, as usual, that the PSFs are normalized in such a way that the sum of all their pixel values is one:

$$\sum_{m,n=0}^{N-1} \mathbf{K}_j(m,n) \;=\; 1 \quad ; \qquad j = 1, 2, ..., p \quad . \tag{4}$$

The problem of image restoration is the problem of estimating the unknown array $\mathbf{f}$, being given the detected images $\mathbf{g}_j$, the backgrounds $\mathbf{b}_j$ and the PSFs $\mathbf{K}_j$. It is well known that it can be reduced to the minimization of a suitable cost functional in the framework of approaches such as regularization theory, maximum likelihood or Bayesian estimation (see, for instance Bertero & Boccacci[4,6]). In the literature there exists a great variety of cost functionals and of iterative algorithms for their minimization, with the possible addition of suitable constraints (for instance, non-negativity of the solution). All these methods have been designed for the deconvolution of a single image and in some cases their extension to the multiple-images case is not trivial. Certainly the software developed for these methods cannot be directly applied to multiple images.

One approach could be to consider some suitable fusion of the multiple images in a single one (see, for instance a recent paper of Vio et al.[21]), so that one can just apply all methods and software developed for the single image case. These methods, however, can modify some important features of the detected images, such as non-negativity, or increase the noise in domains of the u, v plane where information is coming from only one image (this is the result, for instance, of a simple addition of all the images). Therefore in this paper we investigate the extension to the multiple-images case of a general approach which has been recently proposed by Lanteri et al..[16–18] This approach allows the design of iterative algorithms which implement in a quite natural and easy way the constraint of non-negativity; they include, as particular examples, the well-known RL method and the Iterative Space Reconstruction Algorithm (ISRA). Moreover, as we will show in this paper, in the case of multiple images they have a structure which quite naturally suggests iterative algorithms where one performs a cycle over the images so that each iteration has the computational cost of a single image approach. This remark provides algorithms with a feature similar to that of the OSEM method.

In Section 2 we outline the general approach investigated in this paper; in Section 3 we discuss its structure in the multiple-images case and we propose the cyclic algorithms. In Section 4 we provide the information required for applying the general algorithm to the minimization of specific cost functionals and finally in Section 5 we present the new version of the Software Package AIRY. Future developments are discussed in Section 6.

2. THE GENERAL APPROACH

In a series of recent papers Lanteri et al.[16–18] proposed a general approach to the design of iterative methods for the minimization of functionals of the following type:

$$J_\mu(\mathbf{f};\mathbf{g}) \;=\; J_0(\mathbf{f};\mathbf{g}) \;+\; \mu\, J_R(\mathbf{f}) \tag{5}$$

with the additional constraints:

$$\mathbf{f}(m,n) \;\geq\; 0 \qquad , \qquad \sum_{m,n=0}^{N-1} \mathbf{f}(m,n) \;=\; c \;\; . \tag{6}$$

In these equations $J_0(\mathbf{f};\mathbf{g})$ is the functional measuring the discrepancy between the computed images associated to $\mathbf{f}$ and the detected images $\mathbf{g}$ and $J_R(\mathbf{f})$ is a regularization functional while μ is the regularization parameter. As concerns the constraints, the first one is obvious while the second one is a constraint on the total flux of the object.

The basic idea in the approach of Lanteri et al. relies on the following decomposition of the gradient of the functional $J_\mu(\mathbf{f})$:

$$-\nabla J_\mu(\mathbf{f};\mathbf{g}) \;=\; \mathbf{U}_\mu(\mathbf{f};\mathbf{g}) \;-\; \mathbf{V}_\mu(\mathbf{f};\mathbf{g}) \qquad , \tag{7}$$

where $\mathbf{U}_\mu(\mathbf{f};\mathbf{g})$ and $\mathbf{V}_\mu(\mathbf{f};\mathbf{g})$ are positive arrays depending on $\mathbf{f}$. Obviously such a decomposition always exists and is not unique. The applicability of the method requires explicit expressions for the dependence of these arrays on $\mathbf{f}$. Then the general structure of the iterative algorithm, as described in Lanteri et al.,[16] is as follows:

- choose an initial $\mathbf{f^{(0)}}$, satisfying the constraints of Eq. 6 (in general, a uniform image);
- given $\mathbf{f^{(k)}}$ compute:

$$\tilde{\mathbf{f}}^{(k+1)} \;=\; \mathbf{f}^{(k)} + \alpha^{(k)}\mathbf{f}^{(k)}\frac{\mathbf{U}_\mu(\mathbf{f}^{(k)};\mathbf{g}) - \mathbf{V}_\mu(\mathbf{f}^{(k)};\mathbf{g})}{\mathbf{V}_\mu(\mathbf{f}^{(k)};\mathbf{g})} \quad , \quad \tilde{c}^{(k+1)} = \sum_{m,n=0}^{N-1} \tilde{\mathbf{f}}^{(k+1)}(m,n) \;\; , \tag{8}$$

- set:

$$\mathbf{f}^{(k+1)} \;=\; \frac{c}{\tilde{c}^{(k+1)}}\tilde{\mathbf{f}}^{(k+1)} \;\; . \tag{9}$$

In Eq. 8 products and quotients of arrays are defined as pixel-by-pixel products and quotients; $\alpha^{(k)}$ is the step in the descent direction. It can be computed in order to assure both non-negativity of each iterate and convergence of the iterations. Approximate evaluation of $\alpha^{(k)}$, as that given by Armijo rule, usually provides excellent results. However the algorithm takes a very simple form if we take a unit step (i. e. $\alpha^{(k)} = 1$); indeed, in such a case we obtain:

- given $\mathbf{f^{(k)}}$ compute:

$$\tilde{\mathbf{f}}^{(k+1)} \;=\; \mathbf{f}^{(k)}\frac{\mathbf{U}_\mu(\mathbf{f}^{(k)};\mathbf{g})}{\mathbf{V}_\mu(\mathbf{f}^{(k)};\mathbf{g})} \quad , \quad \tilde{c}^{(k+1)} = \sum_{m,n=0}^{N-1} \tilde{\mathbf{f}}^{(k+1)}(m,n) \;\; , \tag{10}$$

- set:

$$\mathbf{f}^{(k+1)} \;=\; \frac{c}{\tilde{c}^{(k+1)}}\tilde{\mathbf{f}}^{(k+1)} \;\; . \tag{11}$$

The convergence of this algorithm is not guaranteed in general, even if it can be proved in some particular cases; nevertheless it has been verified experimentally in all the applications we have considered. The most interesting feature of the algorithm is that *non-negativity of the iterates is automatically satisfied*, as one can easily verify.

In the case of this simplified algorithm the dependence on the regularization parameter can be made explicit by introducing decompositions of the gradients of the functionals $J_0(\mathbf{f};\mathbf{g})$ and $J_R(\mathbf{f})$ similar to that of Eq. 7:

$$-\nabla J_0(\mathbf{f};\mathbf{g}) \;=\; \mathbf{U}_0(\mathbf{f},\mathbf{g}) \;-\; \mathbf{V}_0(\mathbf{f};\mathbf{g}) \qquad , \tag{12}$$

$$-\nabla J_R(\mathbf{f}) \;=\; \mathbf{U}_R(\mathbf{f}) \;-\; \mathbf{V}_R(\mathbf{f}) \qquad ; \tag{13}$$

then it turns out that the algorithm has the following structure:

- given $\mathbf{f^{(k)}}$ compute:

$$\tilde{\mathbf{f}}^{(k+1)} \;=\; \mathbf{f}^{(k)} \frac{\mathbf{U}_0(\mathbf{f}^{(k)};\mathbf{g}) + \mu \mathbf{U}_R(\mathbf{f}^{(k)})}{\mathbf{V}_0(\mathbf{f}^{(k)};\mathbf{g}) + \mu \mathbf{V}_R(\mathbf{f}^{(k)})} \quad , \quad \tilde{c}^{(k+1)} = \sum_{m,n=0}^{N-1} \tilde{\mathbf{f}}^{(k+1)}(m,n) \quad , \tag{14}$$

- set:

$$\mathbf{f}^{(k+1)} \;=\; \frac{c}{\tilde{c}^{(k+1)}} \tilde{\mathbf{f}}^{(k+1)} \quad . \tag{15}$$

which shows the very simple dependence of the algorithm on the regularization parameter μ.

3. APPLICATION TO THE CASE OF MULTIPLE IMAGES

In the case of multiple images the functional $J_0(\mathbf{f};\mathbf{g})$, which measures the misfit between the set of the detected images $\mathbf{g}$ and the set of the computed images associated to $\mathbf{f}$, is, in general, the sum of the functionals corresponding to the different images. In a statistical approach this property follows from the statistical independence of the different images. Therefore we have:

$$J_0(\mathbf{f};\mathbf{g}) \;=\; \sum_{j=1}^{p} J_{0,j}(\mathbf{f};\mathbf{g}_j) \quad . \tag{16}$$

The corresponding decompositions for the functions $\mathbf{U}_0(\mathbf{f};\mathbf{g})$ and $\mathbf{V}_0(\mathbf{f};\mathbf{g})$ are the following:

$$-\nabla J_0(\mathbf{f}) \;=\; \sum_{j=1}^{p} \mathbf{U}_{0,j}(\mathbf{f};\mathbf{g}_j) \;-\; \sum_{j=1}^{p} \mathbf{V}_{0,j}(\mathbf{f},\mathbf{g}_j) \qquad ; \tag{17}$$

if we insert these expression in Eq. 14 and if we remark that the gradient of the regularization term does not depend on j, so that it can be written as the sum of p identical contributions, we find that the complete algorithm takes the following form:

- given $\mathbf{f^{(k)}}$ compute:

$$\tilde{\mathbf{f}}^{(k+1)} \;=\; \mathbf{f}^{(k)} \frac{\sum_{j=1}^{p} \left\{ \mathbf{U}_{0,j}(\mathbf{f}^{(k)};\mathbf{g}_j) + \frac{\mu}{p} \mathbf{U}_R(\mathbf{f}^{(k)}) \right\}}{\sum_{j=1}^{p} \left\{ \mathbf{V}_{0,j}(\mathbf{f}^{(k)};\mathbf{g}_j) + \frac{\mu}{p} \mathbf{V}_R(\mathbf{f}^{(k)}) \right\}} \quad , \quad \tilde{c}^{(k+1)} = \sum_{m,n=0}^{N-1} \tilde{\mathbf{f}}^{(k+1)}(m,n) \quad , \tag{18}$$

- set:

$$\mathbf{f}^{(k+1)} \;=\; \frac{c}{\tilde{c}^{(k+1)}} \tilde{\mathbf{f}}^{(k+1)} \quad . \tag{19}$$

Eq. 18 shows the dependence on both the regularization parameter and the detected images.

The interesting feature of this algorithm is that the numerator and the denominator in Eq. 18 are given by a sum of terms which are just those appearing in the corresponding algorithm for single-image deconvolution. This structure suggests a cyclic-iterative algorithm which is analogous to the OSEM algorithm proposed by Hudson & Larkin[15] in the case of tomography:

	$\mathbf{U}_{0,j}(\mathbf{f};\mathbf{g}_j)$	$\mathbf{V}_{0,j}(\mathbf{f};\mathbf{g}_j)$
Poisson	$A_j^T \frac{\mathbf{g}_j}{A_j\mathbf{f}+\mathbf{b}_j}$	$\mathbf{1}$
Gauss	$A_j^T \mathbf{g}_j$	$A_j^T A_j \mathbf{f} + \mathbf{b}_j$

Table 1. Expression of the functions $\mathbf{U}_{0,j}(\mathbf{f}^{(k)};\mathbf{g}_j)$ and $\mathbf{V}_{0,j}(\mathbf{f}^{(k)};\mathbf{g}_j)$ for the two statistics considered in the text. We denote as $\mathbf{1}$ the array whose entries are all equal to 1.

- given $\mathbf{f}^{(k)}$, set $j = (k+1) \ mod \ p$ and compute:

$$\tilde{\mathbf{f}}^{(k+1)} = \mathbf{f}^{(k)} \frac{\mathbf{U}_{0,j}(\mathbf{f}^{(k)};\mathbf{g}_j) + \frac{\mu}{p} U_R(\mathbf{f}^{(k)})}{\mathbf{V}_{0,j}(\mathbf{f}^{(k)},\mathbf{g}_j) + \frac{\mu}{p}\mathbf{V}_R(\mathbf{f}^{(k)})} \quad ; \quad \tilde{c}^{(k+1)} = \sum_{m,n=0}^{N-1} \tilde{\mathbf{f}}^{(k+1)}(m,n) \quad ; \tag{20}$$

- set:

$$\mathbf{f}^{(k+1)} = \frac{c}{\tilde{c}^{(k+1)}} \tilde{\mathbf{f}}^{(k+1)} \quad . \tag{21}$$

It is obvious that the computational cost of one iteration is just that of one iteration of the same algorithm in the case of a single image. Therefore, if the number of iterations of the algorithms of Eq. 18 and Eq. 20 is approximately the same (a property which has been verified experimentally in several cases), it follows that the algorithm of Eq. 20 provides a computational gain by a factor which is approximately p.

4. EXAMPLES

In this Section we give the tools required for the application of the general approach to specific examples. In particular we consider two kinds of noise statistics, leading to different misfit functionals:

- Poisson Statistics (photon counting) - In this case the functionals $J_{0,j}(\mathbf{f};\mathbf{g}_j)$ are given by:

$$J_{0,j}(\mathbf{f};\mathbf{g}_j) = \sum_{m,n=0}^{N-1} \left\{ \mathbf{g}_j(m,n) \ ln \frac{\mathbf{g}_j(m,n)}{(A_j\mathbf{f})(m,n) + \mathbf{b}_j} + [(A_j\mathbf{f})(m,n) + \mathbf{b}_j - \mathbf{g}_j(m,n)] \right\} \quad ; \tag{22}$$

- Gauss statistics (white noise) - In this case the functionals $J_{0,j}(\mathbf{f};\mathbf{g}_j)$ are given by:

$$J_{0,j}(\mathbf{f};\mathbf{g}_j) = \sum_{m,n=0}^{N-1} |(A_j\mathbf{f})(m,n) + \mathbf{b}_j - \mathbf{g}_j(m,n)|^2 \quad . \tag{23}$$

In Table 1 we give the expressions of the arrays $\mathbf{U}_{0,j}(\mathbf{f}^{(k)};\mathbf{g}_j)$ and $\mathbf{V}_{0,j}(\mathbf{f}^{(k)};\mathbf{g}_j)$ for the two cases. In deriving these formulas we used the normalization of the PSF, as given in Eq. 4, which implies that $A_j^T\mathbf{1} = \mathbf{1}$ and $A_j^T\mathbf{b}_j = \mathbf{b}_j$.

If we insert these expressions into Eq. 18 with $\mu = 0$ we obtain respectively the multi-image extensions of the RL and ISRA algorithms as proposed in Bertero & Boccacci.[5,6] Moreover, by inserting the expressions for the Poisson case into Eq. 20 we re-obtain the OSEM method proposed in Bertero & Boccacci,[5] that we recall now, for completeness:

- given $\mathbf{f}^{(k)}$, set $j = (k+1) \ mod \ p$ and compute:

$$\tilde{\mathbf{f}}^{(k+1)} = \mathbf{f}^{(k)} A_j^T \frac{\mathbf{g}_j}{A_j \mathbf{f}^{(k)} + \mathbf{b}_j} \quad ; \quad \tilde{c}^{(k+1)} = \sum_{m,n=0}^{N-1} \tilde{\mathbf{f}}^{(k+1)}(m,n) \quad ; \tag{24}$$

- set:

$$\mathbf{f}^{(k+1)} = \frac{c}{\tilde{c}^{(k+1)}} \tilde{\mathbf{f}}^{(k+1)} \quad . \tag{25}$$

On the other hand by inserting in the same equation the expressions of the $\mathbf{U}, \mathbf{V}$-arrays for the Gauss case we obtain a new accelerated version of the ISRA algorithm, which can be called OS-ISRA and takes the following form:

- given $\mathbf{f}^{(k)}$, set $j = (k+1) \ mod \ p$ and compute:

$$\tilde{\mathbf{f}}^{(k+1)} = \mathbf{f}^{(k)} \frac{A_j^T \mathbf{g}_j}{A_j^T A_j \mathbf{f} + \mathbf{b}_j} \quad ; \quad \tilde{c}^{(k+1)} = \sum_{m,n=0}^{N-1} \tilde{\mathbf{f}}^{(k+1)}(m,n) \quad ; \tag{26}$$

- set:

$$\mathbf{f}^{(k+1)} = \frac{c}{\tilde{c}^{(k+1)}} \tilde{\mathbf{f}}^{(k+1)} \quad . \tag{27}$$

All these algorithms take into account background addition and flux normalization. As concerns this last point, the constant c can be selected in the following way. In the case of zero background, as shown in Bertero & Boccacci,[5] the total flux of each iterate of the multi-image RL algorithm coincides with the arithmetic mean of the total fluxes of the images $\mathbf{g}_j$. As a consequence of the introduction of the background terms, the iterates of the corresponding RL-algorithm and of its accelerated version OSEM, both given in Bertero & Boccacci,[6] do not have this property. Therefore, it is convenient to constrain the solution to satisfy this property, which implies that the constant c is the arithmetic mean of the total fluxes of the images $\mathbf{g}_j$, after subtraction of the background terms:

$$c = \frac{1}{p} \sum_{j=1}^{N-1} \sum_{m,n=0}^{N-1} \{\mathbf{g}_j(m,n) - \mathbf{b}_j\} \quad . \tag{28}$$

Let us emphasize the fact that the most interesting feature of the approach proposed in this paper is that one can easily insert regularization terms, obtaining new iterative algorithms which can be easily implemented. We mention, among others, Tikhonov regularization and Maximum Entropy both for Gauss and Poisson noise (with the addition of background terms, non-negativity and flux constraint). In Table 2 we give the expressions of the arrays $\mathbf{U}_R(\mathbf{f})$ and $\mathbf{V}_R(\mathbf{f})$ for the following regularization functionals.

- *Tikhonov*

$$J_R(\mathbf{f}) = \frac{1}{2} \sum_{m,n=0}^{N-1} |\mathbf{f}(m,n) - \mathbf{f}^{(R)}(m,n)|^2 \quad , \tag{29}$$

  where $\mathbf{f}^{(R)}(m,n)$ denotes a reference object (in general, a constant array; very often, zero);

- *Smoothing*

$$J_R(\mathbf{f}) = \frac{1}{2} \sum_{m,n=0}^{N-1} |\mathbf{f}(m,n) - (D\mathbf{f})(m,n)|^2 \quad , \tag{30}$$

  where D is a matrix with non-negative entries; such a form applies, for instance, to regularization in terms of discrete Laplacian, which can just be written in the form $\Delta = 4(I - D)$ (I being the identity matrix);

$J_R(f)$	$\mathbf{U}_R(\mathbf{f})$	$\mathbf{V}_R(\mathbf{f})$
$Tikhonov$	$\mathbf{f}^{(R)}$	$\mathbf{f}$
$Smoothing$	$(D+D^T)\mathbf{f}$	$(I+D^TD)\mathbf{f}$
$Entropy$	$-ln(\frac{1}{c}\mathbf{f})$	$1-ln(\frac{1}{c}\mathbf{f}^{(R)})$

Table 2. Expressions of the functions $\mathbf{U}_R(\mathbf{f})$ and $\mathbf{V}_R(\mathbf{f})$ in the case of Tikhonov, Smoothing and Entropy regularization. In the case of Entropy the log of an array is defined pixel by pixel; moreover the flux normalization is taken into account, c being the flux constant defined in Eq. 28.

- *Entropy*

$$J_R(\mathbf{f}) \;=\; \sum_{m,n=0}^{N-1} \mathbf{f}(m,n)\; ln\frac{\mathbf{f}(m,n)}{\mathbf{f}^{(R)}(m,n)} \quad , \tag{31}$$

  where, again, $\mathbf{f}^{(R)}(m,n)$ denotes a reference object.

If we combine the arrays of Table 1 and Table 2 we obtain several iterative algorithms for different types of regularization both in the Poisson and in the Gauss case. As an example we give the regularization of the RL algorithm with the Tikhonov functional, assuming $\mathbf{f}^{(R)}=0$; the resulting algorithm is a slightly modified version of OSEM:

- given $\mathbf{f}^{(k)}$, set $j \;=\; (k+1) \;\; mod\; p$ and compute:

$$\tilde{\mathbf{f}}^{(k+1)} \;=\; \frac{\mathbf{f}^{(k)}}{\mathbf{1}+\frac{\mu}{p}\mathbf{f}^{(k)}} A_j^T \frac{\mathbf{g}_j}{A_j\mathbf{f}^{(k)}+\mathbf{b}_j} \quad ; \quad \tilde{c}^{(k+1)} = \sum_{m,n=0}^{N-1} \tilde{\mathbf{f}}^{(k+1)}(m,n) \quad ; \tag{32}$$

- set:

$$\mathbf{f}^{(k+1)} \;=\; \frac{c}{\tilde{c}^{(k+1)}} \tilde{\mathbf{f}}^{(k+1)} \quad . \tag{33}$$

In a similar way we obtain the Tikhonov regularization of OS-ISRA:

- given $\mathbf{f}^{(k)}$, set $j \;=\; (k+1) \;\; mod\; p$ and compute:

$$\tilde{\mathbf{f}}^{(k+1)} \;=\; \mathbf{f}^{(k)} \frac{A_j^T\mathbf{g}_j}{A_j^TA_j\mathbf{f}^{(k)}+\mathbf{b}_j+\frac{\mu}{p}\mathbf{f}^{(k)}} \quad ; \quad \tilde{c}^{(k+1)} = \sum_{m,n=0}^{N-1} \tilde{\mathbf{f}}^{(k+1)}(m,n) \quad ; \tag{34}$$

- set:

$$\mathbf{f}^{(k+1)} \;=\; \frac{c}{\tilde{c}^{(k+1)}} \tilde{\mathbf{f}}^{(k+1)} \quad . \tag{35}$$

Module	Purpose
Data simulation modules	
`OBJ - OBJect definition`	-to define the object characteristics
`CNV - object-PSF CoNVolution`	-to perform convolution
`ADN - ADd Noise to image`	-to add background, read-out and Poisson noise
Data processing modules	
`PRE - PRE-processing`	-to perform image pre-processing
`DEC - DEConvolution process`	-to perform deconvolution (OS-EM method)
Data analysis modules	
`ANB - ANalysis of Binary`	-to analyse reconstructed images of binary objects
`FSM - Find Star Module`	-to detect stars in the reconstructed images
Other modules and utilities	
`RFT - Read FiTs file format`	-to read FITS images
`WFT - Write FiTs file format`	-to write FITS images
`RSC - Restore im. Struct. Cubes`	-to restore image structure cubes (XDR or FITS format)
`SIM - Save IMage struct.`	-to save image structure cubes (XDR or FITS format)
`DSP - data DiSPlay`	-to display images

Table 3. Descriptive list of the modules of the Software Package AIRY (as in version 2.1).

5. THE SOFTWARE PACKAGE AIRY

As outlined in the previous Sections, the general approach presented in this paper contains, as a particular case, the OSEM method we have proposed and validated for the deconvolution of LBT images. This method is implemented in the Software Package AIRY,[11,12] designed and developed for simulating optical and near-infrared interferometric observations and/or performing subsequent image restoration/deconvolution. It was conceived for application to LBT, but can also be used for future interferometers of the same class.

AIRY can be used together with the CAOS (Code for Adaptive Optics Systems) Application Builder,[13,14] which provides a graphical programming environment where elementary building blocks (modules) can be assembled together to create a simulation program (project). A user can build a project by putting together the needed modules and specifying the data flow between each block. When the simulation has been defined the IDL code which implements the program is automatically generated.

AIRY, like CAOS, is written in IDL and consists of a set of specific modules, each representing a particular task. Those implemented in the present version are described in Table 3. Like in CAOS a user can build a project by linking the modules together and setting their parameters through their own graphical user interfaces (GUIs); moreover she/he can easily add new modules for her/his particular needs. In the present version of AIRY only two types of input/output are defined: the "object" type and the "image" type, but almost all modules have inputs and outputs of the type "image". Moreover any module can have a number of inputs/outputs varying between zero and two and the output of each module can be linked to the compatible input of another one.

As concerns deconvolution, AIRY includes both an implementation of the extension of the RL-method to multiple images and its accelerated version known as the OSEM method.

Figure 1 shows a typical simulation project in terms of the modules of the Software Package AIRY. The astrophysical object is defined in great detail within the module OBJ, while the simulated PSFs are read from the disk within the module RFT; convolution is then performed by the module CNV. Next the module ADN adds the background to the images coming from CNV and corrupts the results with different kinds of noise; the data are preprocessed (sky-background estimation, etc.) within the module PRE, and finally the deconvolution is performed by the module DEC. The quantities of interest (original object, PSF, reconstructed image) are displayed at each iteration of the deconvolution method by means of the module DSP.

The Software Package AIRY is available upon request and subscription to the dedicated mailing-list by visiting the web site `http://www.dirac.disi.unige.it`. The CAOS Application Builder graphical environment is directly downloadable from the dedicated web site `http://www.arcetri.astro.it/caos`.

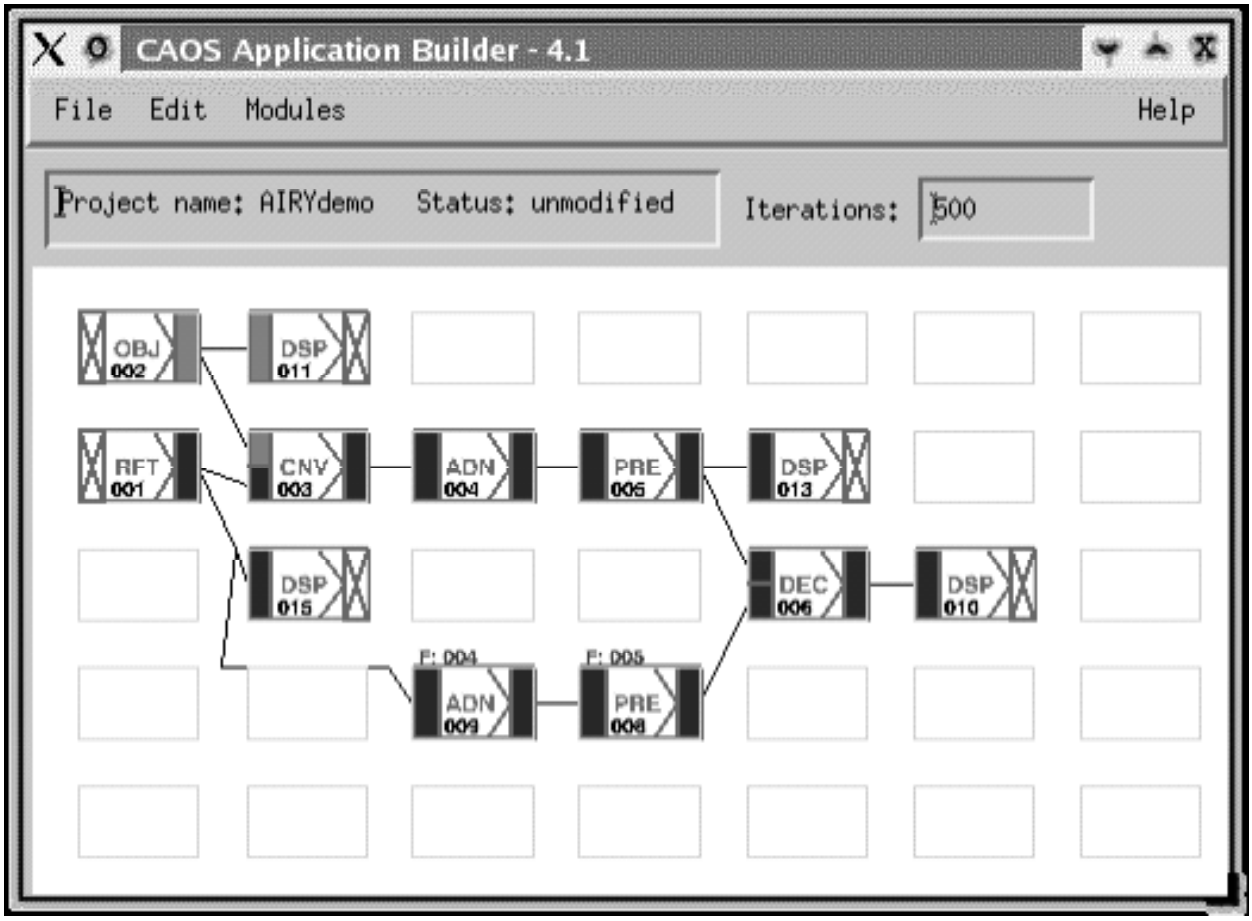

Figure 1. Example of a simulation project designed by using the modules of the Software Package AIRY, version 2.1, within the CAOS Application Builder programming environment, version 4.1.

5.1. Current Status

Version 2.0 of the Software Package AIRY has been distributed among the community of the registered users in December 2003. With respect to the previously distributed Version (1.0), the main structural difference concerns the compatibility with recent versions of the CAOS Application Builder (from 4.0 on), which allow to have different *Software Packages* working together, and a number of new features which make easier the building of a simulation project.

Version 2.0 contains the same number of modules as Version 1.0, but several modifications and upgrades have been introduced, mainly in the modules DEC, OBJ and PRE. In particular, the module PRE contains now a new kind of estimation of the sky background, which is provided by the user and hence is not dependent on a particular algorithm; moreover we added the possibility of subtracting or not subtracting the background from the images, for different purposes of pre-processing studies. We have also introduced some important modifications in the module DEC: 1 - the standard RL method for multiple-images deconvolution, so that the user can choose between this method and its accelerated version OSEM; 2 - an algorithm for super-resolving compact objects such as a binary system with an angular separation smaller than the angular resolution of the telescope.

We have now completed Version 2.1 where the most important modifications concern the module ADN. With this module it is now possible to consider also the dark-current noise, and to have control on the seeds for the

generation of all types of noise (photon noise, read-out noise, and dark-current noise). Moreover in the module DEC we have introduced an additional acceleration of OSEM (which will be discussed in a forthcoming paper)[2] which allows for a reduction by a factor $\sim$2 in computational time.

The utility and the potentiality of the Software Package AIRY have been widely demonstrated by a number of scientific studies whose main results are recalled in the next sub-section.

5.2. Applications

We have used our tool for the study of some interesting features and limitations of the imaging by a Fizeau interferometer such as LBT. In particular we have investigated the effect of angular coverage and angular smearing, which depend on the declination of the observed object, and the effect of partial adaptive optics (AO) correction, asuming two different schemes (classical AO and multiconjugate AO).[8] Moreover we have tested the performance of OSEM for the super-resolution of binary systems as well as the performance of some new methods of image deconvolution.[1]

All our studies were performed by simulating high-resolution interferometric observations of an astronomical object and by comparing the scientific parameters of the restored object with those of the original one. In particular the study was performed on different types of objects, from binary stars to diffuse objects, including a combined one with a very high dynamic range.

The limitation in angular coverage is correlated with the problem of angular smearing due to the time-averaging of the interferometric images. It was investigated for relevant cases depending on the declination of the observed object. Results are encouraging even in the case of incomplete coverage, and the angular coverage and smearing seem, at the light of the simulations performed, not to affect significantly the large structure of a diffuse object nor the retrieval of the parameters of a close binary star, but only the convergence rate of the iterative method.[8,9]

Partial AO-correction can result in a wide range of image quality, but can also create significant differences within a same field-of-view, especially between a suitable reference star to be used for post-observation multiple deconvolution and the observed object. We investigated both the effect of the space-variance of the AO-corrected PSF,[8] and that of the global quality of the AO correction.[9] It turns out that uniformity over the field is a main issue for this observational technique, even at the cost of a lower average Strehl ratio. Such a condition, which may be satisfied by a multiconjugate AO system like LN, is also required by the model of image formation assumed in the deconvolution method and implies that the PSF obtained from a reference star provides a good approximation of the PSF which must be used in the process of image restoration.

The method for super-resolving compact objects[1] is based on a simple modification of the RL or OSEM method and in general consists of 3 steps: the first one requires a large number of RL/OSEM iterations, which are used to estimate the domain of the unresolved object; the second one is a RL/OSEM-restoration inizialized with the mask of the domain, as estimated in step 1. These two steps provide a correct estimate of the positions of the two stars while their magnitudes can be obtained in a possible 3rd step by solving a simple least-squares problem. The first two steps are included in our module DEC. In the second step it is possible to choose, within the DEC interface, the image and the mask used to initialized the method. The mask is an image with values 0 or 1, and three types of mask are considered: the 1st one is based on a percentage of the image maximum, the 2nd one is a circular mask, and the 3rd one is a user-defined mask.

Figure 2 shows a typical set of PSFs routinely used in our simulation studies, and corresponding to the three (relative) parallactic angles of $0, 60$ and 120 deg (corresponding respectively to the three successive columns of the figure). The effect of partial AO-correction is shown in the first row: different Strehl ratios are obtained for the three parallactic angles, due to the simple fact that they are obtained at three different moments of an observing night, and hence with three different atmospheric turbulence conditions (implying three different sets of tuning parameters for the AO system – see Carbillet et al.[10]).

The different values of the Strehl ratio (SR) are 52% for the PSF at 0 deg (left), 87% for the PSF at 60 deg (middle), and 79% for the PSF at 120 deg (right). The difference in terms of PSF morphology are evident from this figure, where we go from the more speckle-ized appearance of the PSF at 52% of SR, to the "cleaner" aspect of the PSF at 87% of SR. From the point-of-view of angular smearing, the differences between the three PSFs is

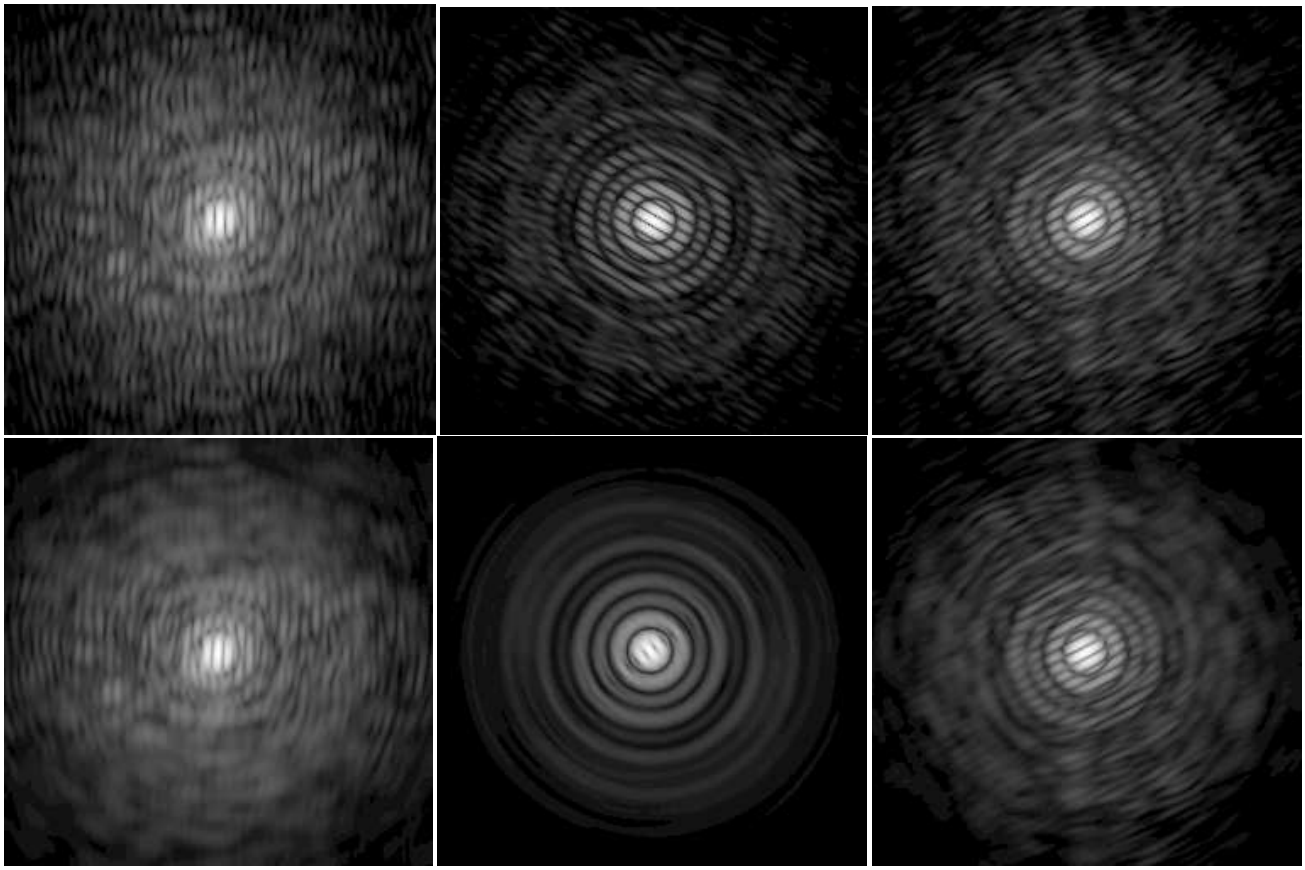

Figure 2. A typical set of simulated AO-corrected and angular-smeared PSFs used within our various LN image restoration studies (logarithmic scale). See the text for details.

also very evident (see the second row of the figure): while the first and the third PSFs (from the left) are smeared over an angle of about 5 deg, the second one is smeared over a much broader angle, of about 45 deg. This effect of rotation of the baseline during the integration time is obtained by adding snapshot PSFs for different angles, with a step of 1 deg.

6. CONCLUDING REMARKS

Imaging with LN will require routinely the use of multiple-images deconvolution methods because a unique high-resolution image must be extracted from different interferometric observations of the same target, obtained with different orientations of the baseline with respect to the sky. If one takes into account the complexity of astronomical images, two kinds of methods will be required. The first consists of "quick-look methods", namely computationally efficient methods which permit to have a rapid overview of the object being observed. The second consists of "ad-hoc methods" designed for a particular class of objects and as accurate as possible for that particular class. The general approach outlined in this paper provides a unifying framework for the development of both types of methods.

As for the quick-look methods, we have developed OSEM and for the future our program is to increase the efficiency of the OSEM algorithm by means of techniques which allow a reduction of the number of iterations.[2]

As for the ad-hoc methods, we will first focus on the case of objects with a high dynamic range. Indeed, high angular resolution coupled with high dynamic range capabilities will permit to obtain, for example in the field of T Tauri binaries, detailed images of the circumbinary disks. Most of the theoretical predictions about the structure of circumbinary disks and their interactions with their central binaries could be probed, as for

example the replenishment streams feeding one of the star's circumstellar disk.[3] Application to protoplanetary disk observations, as well as detection of giant planets around white dwarfs[7] or nearby young stars, are also envisaged.

In the framework of the super-resolution method we have developed,[1] we are currently studying its practical application. In particular, the data oversampling, required for this analysis, can probably be obtained numerically. Note that we expect with this method to reach unprecedented angular resolutions. In fact, with a four times gain in angular resolution, we could obtain with LBT an image with a resolution of a 100 m telescope. This is equivalent to the resolution of the VLTI, but with the advantage of obtaining images of the possibly complex picture coming into play for example in star formation. Such a resolution corresponds to less than 1 AU (even in K band) at the distance of the nearest and best-studied star forming dark clouds and stellar associations (about 140 pc). This is especially useful in order to bridge the gap in period/separation between visual T Tauri binaries and spectroscopic ones. Moreover, this is of fundamental importance in order to correct for binary luminosities, ages, and masses of previously thought single objects.[19] While detailed studies of circumstellar disks structure and properties will mostly benefit from higher resolution observations (with the VLTI or future interferometers of the same kind), detection and low-resolution images could be attempted with such a technique. A survey of multiplicity among T Tauri stars will also take advantage of the fainter limiting magnitude provided by LBT with respect to VLTI.

REFERENCES

1. Anconelli B., Bertero M., Boccacci P. & Carbillet M., 2004, submitted to A&A
2. Anconelli B., Bertero M., Boccacci P., Carbillet M., & Lanteri H., 2004, in preparation
3. Artymowicz P. & Lubow S. H., 1996, ApJ, 467, L77
4. Bertero M., & Boccacci P., 1998, *Introduction to Inverse Problems in Imaging* (IOP Publishing, Bristol)
5. Bertero M., & Boccacci P., 2000a, A&AS, 144, 181
6. Bertero M., & Boccacci P., 2000b, A&AS, 147, 323
7. Burleigh M. R., Clarke F. J., & Hodgkin S. T., 2002, MNRAS, 331, 41
8. Carbillet M., Correia S., Boccacci P., & Bertero M., 2002, A&A, 387, 744
9. Carbillet M., Correia S., Bertero M., & Boccacci P., 2003a, SPIE Proc. **4840**, W. A. Traub Ed., 444
10. Carbillet M., Vérinaud C., Esposito S., Riccardi A., Puglisi A., Femenía B., Fini L., 2003b, SPIE Proc. **4839**, P. L. Wizinowich & D. Bonaccini Eds., 131
11. Correia S., Carbillet M., Boccacci P., Bertero M., & Fini L., 2002, A&A, 387, 733
12. Correia S., Carbillet M., Fini L., Boccacci P., Bertero M., Vallenari A., Richichi A., & Barbati M., 2001, ASP Conf. Ser. **238**, F. A. Primini & F. R. Harnden Eds., 404
13. Fini L., Carbillet M., Riccardi A., 2001, ASP Conf. Ser. **238**, F. A. Primini & F. R. Harnden Eds., 253
14. Fini L. & Carbillet M., 2003, ASP Conf. Ser. **295**, H. E. Payne, R. I. Jedrzejewski, & R. N. Hook Eds., 347
15. Hudson H. M., & Larkin R. S., 1994, IEEE Trans. Med. Im., 13, 601
16. Lanteri H., Roche M., Cuevas O., & Aime C., 2001, Signal Processing, 81, 945
17. Lanteri H., Roche M., & Aime C. 2002, Inverse Problems, 18, 1397
18. Lanteri H., Roche M., Gaucherel P., & Aime C., 2002, Signal Processing, 82, 1481
19. Simon M., Ghez A., Leinert Ch., 1993, ApJ, 408, L33
20. Snyder D. L., Hammoud A. M., & White R. L., 1993, JOSA, A10, 1014
21. Vio R., Nagy J., Tenorio L., & Wamsteker W., 2004, A&A, 416, 404