

# FOCUS: a deconvolution method based on algorithmic complexity

C. Delgado

Instituto de Astrofísica de Canarias, Avda. Vía Láctea s/n, 38205 La Laguna, Tenerife, Spain  
e-mail: delgadom@iac.es

Received 3 August 2005 / Accepted 11 March 2006

## ABSTRACT

A new method for improving the resolution of images is presented. It is based on Occam's razor principle implemented using algorithmic complexity arguments. The performance of the method is illustrated using artificial and real test data.

**Key words.** methods: data analysis – techniques: image processing – methods: numerical

## 1. Introduction

Any recorded image is blurred whenever the instrument used to obtain it has a finite resolving power: for example, the image of a point source seen through a telescope has an angular size that is inversely proportional to the diameter of the primary mirror. If the instrument is ground-based, the image is also degraded by the turbulent motions in the earth's atmosphere. Most of the scientific information can be extracted from these blurred images, so further effort to improve its spatial resolution is not necessary. However, it is desirable for some studies to improve it as much as possible. Currently much effort is being devoted to developing methods of obtaining this improvement either by introducing new observing techniques (e.g. interferometry or adaptive optics, Lená 1988) or by a subsequent numerical deconvolution of the image. The method we introduce in this paper belongs to the latter class.

An observed image may usually be mathematically expressed as a convolution of the original light distribution with the point spread function (PSF), including atmospheric perturbation (seeing) if relevant. The resulting distribution is subsequently sampled by the telescope camera, so that the measured light distribution becomes

$$\mathbf{d}_i = \sum_{j=1}^N \mathbf{R}_i^j \mathbf{t}_j + \mathbf{n}_i \quad (1)$$

where  $N$  is the number of sampling points in the observing instrument, vector  $\mathbf{d}$  is the measured distribution of photons,  $\mathbf{t}$  is the true light distribution, and  $\mathbf{n}_i$  the statistical noise at pixel  $i$ . The matrix element  $\mathbf{R}_i^j$  is the value at the sampling point  $i$  of the PSF centered in  $j$ . Notice that for the usual CCD images, the term  $\mathbf{n}_i$  has two contributions at least: the noise due to the fluctuation of the background, which can be considered Gaussian, and the statistical fluctuation of the number of detected photons, which follows Poisson statistics.

Given  $\mathbf{d}$  and the PSF, the aim of deconvolution is to estimate the original light distribution  $\mathbf{t}$ . This problem does not have a unique solution, due in part to the ubiquitous noise. Therefore, regularization techniques have to be used in order to select a plausible one amongst the family of all the solutions that are statistically compatible with the observed image.

## 2. The method

A wide variety of deconvolution methods have been proposed, each depending on the way a particular solution is chosen. Among those that are quite popular are the Richardson-Lucy (RL) (Lucy 1974; Richardson 1972) and the maximum entropy method (MEM) (Reiter 1990; Sault 1990). The former relies on an iterative approximation to the maximum likelihood solution assuming Poisson distributed noise, whereas the regularization is carried out by choosing the number of iterations according to some criteria. The MEM method chooses the solution with maximum entropy among those that maximize the likelihood weighted by a regularization parameter. Usually these regularization techniques suffer from problems due to correlations between pixels, especially in the presence of bright sources over a smooth background. These problems can be largely suppressed if the number of degrees of freedom for the solution of the first equation is reduced, as in the Pixon method (Puetter & Yahil 1999).

Normal everyday science is implicitly ruled by Occam's razor; that is, among a set of models to describe a single process, the simplest one explaining the data is preferred. Our deconvolution method applies the same principle to choose the solution to Eq. (1). In order to achieve this, it is necessary to measure the simplicity of any test solution. To this end we used the Kolmogorov algorithmic complexity (AC) (Kolmogorov 1965), also known as the algorithmic information content, defined as the length of the shorter binary string that fully describes a given object, in our case the measured image<sup>1</sup>. Among others, this quantity offers one property that makes it very attractive for our purpose. Given the AC  $K(\mathbf{d}, \mathbf{m})$  for the data  $\mathbf{d}$  and the model  $\mathbf{m}$ , then except for a normalization constant independent of the data, the best lower bound of the probability for that dataset is given by (Levin 1974; Li & Vitanyi 1997)

$$P(\mathbf{d}, \mathbf{m}_{\text{best}}) \sim e^{-K(\mathbf{d}, \mathbf{m}_{\text{best}})} \quad (2)$$

where  $\mathbf{m}_{\text{best}}$  is the model that minimizes  $K(\mathbf{d}, \mathbf{m})$ .

<sup>1</sup> Usually this length is measured in *bits*, such that the number of different strings is  $2^l$ . However in this paper we measure it in *nats*, the number of different strings being  $e^l$ . With this convention the base of any logarithm in any equation is assumed to be the natural one.

It can be shown that  $K(\mathbf{d}, \mathbf{m})$  cannot be computed. Fortunately, using the additivity properties of the AC and the Shannon-Fano code for data compression, the following bound from above can be derived (Li & Vitanyi 1997) as

$$\log P(\mathbf{d}, \mathbf{m}) = -K(\mathbf{d}, \mathbf{m}) + c1 \leq \log P(\mathbf{d}|\mathbf{m}) - K(\mathbf{m}) + c2 \quad (3)$$

where  $P(\mathbf{d}|\mathbf{m})$  is the likelihood function (probability of the data given the model),  $K(\mathbf{m})$  the AC of the model, and  $c1$  and  $c2$  are constants that only depend on the model.

Obviously we are interested in maximizing Eq. (3), such that the optimal model  $\mathbf{m}_{\text{best}}$  gives the best estimate to the solution of the deconvolution problem. In this maximization the term  $-K(\mathbf{m})$  acts as a model prior constructed from the algorithmic information arguments. As in the case of  $K(\mathbf{m}, \mathbf{d})$ , it cannot be computed (Li & Vitanyi 1997). However, it is possible to estimate a lower bound if the class of models under consideration is constrained. In our case we consider  $\mathbf{m}$  can be any histogram obtained by sampling a continuous distribution with the same sampling points as the data.

### 2.1. Estimate of the likelihood function

In order to find closed analytic expressions for  $P(\mathbf{d}|\mathbf{m})$  and  $K(\mathbf{m})$ , we assume that  $\mathbf{d}_i$  and  $\mathbf{m}_i$  have been scaled such that they give the number of photons in pixel  $i$  after and before the convolution, respectively. With these assumptions the expected number of measured photons at pixel  $i$ , given  $\mathbf{m}$ , is

$$\langle \mathbf{d} \rangle_i = \sum_{j=1}^N \mathbf{R}_i^j \mathbf{m}_j. \quad (4)$$

Furthermore if we approximate that the image formation in the absence of background noise is described well by a Poisson process, the likelihood function can be written as

$$\log P(\mathbf{d}|\mathbf{m}) = \sum_{i=1}^N \log \text{Pois}(\mathbf{d}_i | \langle \mathbf{d} \rangle_i) \quad (5)$$

where  $\text{Pois}(x | \langle \mathbf{d} \rangle_i)$  is the Poisson probability distribution of mean  $\langle \mathbf{d} \rangle_i$  evaluated at  $x$ .

To take the background into account noise in the evaluation of the likelihood function, we consider that for any pixel it is described well by a Gaussian distribution with zero mean and a known standard deviation. This noise is included in the likelihood function by convolving it with each contributing Poisson distribution in Eq. (5). The result is approximated well by a Gaussian distribution with mean  $\langle \mathbf{d} \rangle_i$  and standard deviation  $\sqrt{\langle \mathbf{d} \rangle_i + \sigma_i^2}$  for pixel  $i$ , where  $\sigma_i$  is the noise standard deviation in this pixel. In consequence the final log likelihood function reads

$$\begin{aligned} \log P(\mathbf{d}|\mathbf{m}) \simeq & -\frac{1}{2} \sum_{i=1}^N \left[ \frac{(\mathbf{d}_i - \sum_{j=1}^N \mathbf{R}_i^j \mathbf{m}_j)^2}{\sigma_i^2 + \sum_{j=1}^N \mathbf{R}_i^j \mathbf{m}_j} \right. \\ & \left. + \log \left( 2\pi \left( \sigma_i^2 + \sum_{j=1}^N \mathbf{R}_i^j \mathbf{m}_j \right) \right) \right]. \end{aligned} \quad (6)$$

The obtained function differs from the one in other deconvolution methods, which generally use a Gaussian likelihood (Magain et al. 1998; Mugnier et al. 2004; Gull & Skilling 1991). The main difference is the dependence on  $\langle \mathbf{d} \rangle_i$  of the total standard deviation for pixel  $i$ , which gets replaced by  $\mathbf{d}_i$  in

the Gaussian case. The conditions for the validity of this approximation can be easily obtained from Eq. (6). If we define  $\delta_i = \mathbf{d}_i - \langle \mathbf{d} \rangle_i$ , the maximum condition up to the first order in  $\delta_i$  reads

$$-\frac{\partial \log P(\mathbf{d}|\mathbf{m})}{\partial \delta_i} = \frac{\delta_i}{\sigma_i^2 + \mathbf{d}_i} - \frac{1}{2} \frac{1}{\sigma_i^2 + \mathbf{d}_i} = 0. \quad (7)$$

The solution to this equation is close to the Gaussian likelihood maximum if  $\sigma_i + \mathbf{d}_i \gg 1$ . Although this condition is fulfilled in most practical cases, we will adopt Eq. (6) in our algorithm, as it is the appropriate one for  $\mathbf{m}$  far from the maximum and for pixels with a low number of detected photons.

### 2.2. Estimate of $K(\mathbf{m})$

In order to estimate the  $K(\mathbf{m})$ , it is necessary to introduce some concepts. Let us start by obtaining the amount of algorithmic information needed to give a single number. Then these results will be used to compute the needed quantity.

#### 2.2.1. Algorithmic information for a single number

It can be shown that for almost any integer  $n > 0$ , the amount of algorithmic information necessary to represent it is approximately equal to the number of *bits* needed to write  $n$  in binary notation (Li & Vitanyi 1997; Risannen 1983):

$$K_2(n) = \text{ceil}(\log_2 n).$$

The subindex 2 reminds that we are measuring in *bits* instead of *nats*, and the function  $\text{ceil}(x)$  gives the closer integer to  $x$  from above.  $K(x)$  is obtained by simply replacing the base 2 logarithm by the natural one. A good approximation from a numerical point of view, is obtained by ignoring the *ceil* function, resulting in

$$K(n) \simeq \log n \quad (8)$$

where the information is measured in *nats*.

In order to extend this result to any integer number, it is necessary to add the sign information and to deal with the zero in the binary number representation. The resulting  $K(n)$  function is

$$K(n) \simeq \begin{cases} \log(|n - n_0| + 1) + \log 2 & n \leq 0 \\ \log(|n - n_0|) + \log 2 & n > 0 \end{cases}$$

or the same expression but swapping the conditions and the +1 within the logarithmic function. The  $\log 2$  term in this expression accounts for the information needed to give the sign of the integer number. On the other hand,  $n_0$  is the position of the *zero-informative value*, that is, the value for which the number  $n$  does not need to be represented. Clearly if  $|n - n_0|$  is large enough, we can approximate

$$K(n) \simeq \frac{1}{2} \log((n - n_0)^2 + 1) + \log 2. \quad (9)$$

The +1 within the logarithm function has been included to obtain the correct result for  $n - n_0 = 0$ , such that this approximation remains finite for any  $n$ .

The extension of Eq. (9) to a real number  $x$  is easily achieved if the precision  $s$  needed for representing it in a given context is known a priori. Then we can compute  $K^s(x) = K(n_x) + K(s)$ , where  $n_x$  is the integer that best approximates  $x/s$ , and  $K(s)$  is a constant independent of  $x$ .

Before using these quantities to estimate  $K(\mathbf{m})$ , some words are needed about the *zero-informative value*. Although there is an arbitrary number, when describing a model it depends on our a priori knowledge or our assumptions about it. This is because if a parameter has this value, the model is equivalent to itself without the corresponding degree of freedom from the point of view of the amount of algorithmic information. Together with the precision, this quantity has to be chosen carefully when obtaining a good estimate of  $K(\mathbf{m})$ .

### 2.2.2. From $K(x)$ to $K(\mathbf{m})$

First of all let us recall that any correct estimate of  $K(\mathbf{m})$  is an upper bound of the true one, so any a priori knowledge of the model can be used to improve it.

In the absence of further information, model  $\mathbf{m}$  is fully described by giving the number of photons at each pixel. On physical grounds, we do not expect the needed precision to be less than a single photon, so we adopt this as the scale. Therefore the algorithmic information is

$$K(\mathbf{m}) \simeq c_1 + \sum_{i=1}^N K(\mathbf{m}_i) \simeq c_2 + \sum_{i=1}^N \frac{1}{2} \log(m_i^2 + 1) \quad (10)$$

where  $c_1$  and  $c_2$  are constants. It must be stressed that with this representation the *zero-informative value* is at  $\mathbf{m}_i = 0$ . Equation (10) has a simple interpretation in terms of the relative entropy or Kullback-Leibler distance (Kullback & Leibler 1951): using Eq. (8) for  $\mathbf{m}_i > 0$  we can be write

$$\sum_{i=1}^N K(\mathbf{m}_i) = -N \sum_{i=1}^N c_i \log \frac{c_i}{\mathbf{m}_i/M} - N \log \frac{N}{M} \quad (11)$$

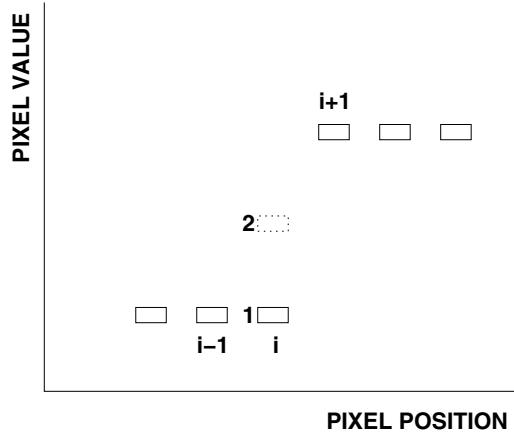
where  $M = \sum_{i=1}^N \mathbf{m}_i$  and  $c_i = 1/N$ . The first term on the right hand side of this expression is proportional to the the relative entropy of the uniform distribution with respect to the distribution of photons  $\mathbf{m}$  normalized to unity. Clearly this approach is close to the standard maximum entropy regularization with a uniform prior image in the form given by Ables (1974), Wernecke (1977), and Wernecke & D'Addario (1977). Furthermore, as Eq. (10) gives an upper bound to the true  $K(\mathbf{m})$ , it is feasible to improve it by including an a priori unknown multiplicative regularization parameter  $\alpha \lesssim 1$ .

A more natural way of improving the bound within the algorithmic information context follows from the use of a priori information on the distribution of photons. As a simple example of how this can be done, assume that it is known that  $\mathbf{m}_i$  is around a fixed value. Then we can shift the *zero-informative value* to it to obtain a smaller, therefore closer,  $K(\mathbf{m})$  estimate.

In the same spirit as (Molina et al. 1992) and (Magain et al. 1998), we conjecture the best model image is smooth or piecewise smooth, so Eq. (10) is discarded in favour of an estimated model algorithmic complexity involving differences between neighbouring pixels. From this simple consideration follows the next estimate

$$K_{\text{mean}}(\mathbf{m}) = \sum_{i=1}^N K^{s_i} \left( \mathbf{m}_i - \frac{1}{Nb(i)} \sum_{j \in \text{neighbours}(i)} \mathbf{m}_j \right) \quad (12)$$

where  $s_i$  is, according to the previous section, the fixed scale to compute the real number contribution of pixel  $i$ ,  $\text{neighbours}(i)$  is the set of pixels that are neighbours of  $i$ , and  $Nb(i)$  is the number of neighbours of that pixel. Interpretation of  $K_{\text{mean}}(\mathbf{m})$  is simple: within our conjecture, the amount of information needed to



**Fig. 1.** One-dimensional example of the computation of  $K_{\text{mean}}(i)$  on the edge of a smooth region for pixel  $i$ . The neighbours are  $i-1$  and  $i+1$ . The value of a pixel is represented by the continuous line boxes. The dotted line one represents an effective pixel value (see text).

describe a pixel image is the one needed to give the difference of its value with the mean of its neighbourhood. Intuitively this estimate works correctly within smooth regions in the image. However at the edges of one of these regions, the value of the information becomes too high, as exemplified in Fig. 1; according to it, the pixel  $i$  value is **1**, so the amount of information needed to describe it should be  $K^{s_i}(\mathbf{m}_i - \mathbf{m}_{i-1})$ . Nevertheless, a  $K_1(\mathbf{m})$  estimate results in the algorithmic information as if the pixel value were at position **2**. As a consequence of this behaviour, its use in Eq. (3) will tend to oversmooth edges and point like sources.

If the closest neighbour  $j$  to pixel  $i$  in value for the best model were known, we could avoid the edge problem by using directly  $K^{s_i}(\mathbf{m}_i - \mathbf{m}_j)$ . This can be approximated by a robust estimate  $e(\mathbf{m}_i)$  of  $\mathbf{m}_i$  (Rey 1983; Press et al. 1992a), estimating algorithmic information of the model as

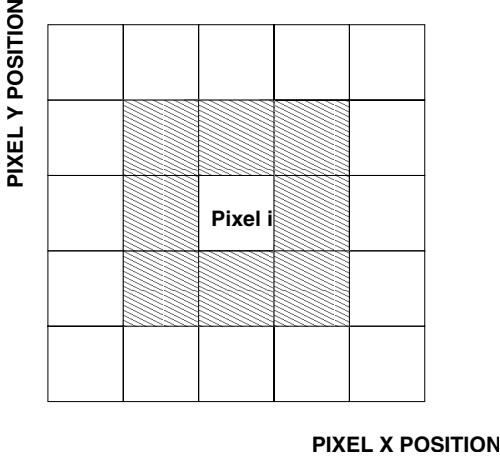
$$K_{\text{robust}}(\mathbf{m}) = \sum_{i=1}^N K^{s_i}(\mathbf{m}_i - e(\mathbf{m}_i)). \quad (13)$$

Unfortunately this approach has the disadvantage of lacking a closed analytic form. Between this later estimate and  $K_{\text{mean}}$  is the following expression, that we take as our final analytic approximation for  $K(\mathbf{m})$ :

$$K_{\text{Cauchy}}(\mathbf{m}) = \sum_{i=1}^N \frac{1}{Nb(i)} \sum_{j \in \text{neighbours}(i)} K_{s_i}(\mathbf{m}_i - \mathbf{m}_j), \quad (14)$$

which can be interpreted as the mean information needed to describe  $\mathbf{m}_i$  from one of its neighbours. The application of this expression on the example of Fig. 1 shows that  $K_{\text{Cauchy}}(\mathbf{m})$  leads to a better estimate of the algorithmic information in the edge compared with Eq. (12). Due to its form, we expect it to be as good as  $K_{\text{mean}}(\mathbf{m})$  within smooth regions.

The estimate  $K_{\text{Cauchy}}(\mathbf{m})$  has some remarkable properties. It is invariant under a global shift of the values  $\mathbf{m}_i$ , so it does not bias the global photometry of the deconvolution. So, it is invariant under scale transformation if the values  $s_i$  are scaled accordingly. On the other hand, for any reasonable dependence of  $s_i$  with the number of detected photons  $M$ ,  $K_{\text{Cauchy}}(\mathbf{m})$  grows as much as  $\log M$ , whereas the log likelihood grows as  $M$ . Therefore in the limit  $M \rightarrow \infty$ , the maximum likelihood deconvolution solution is recovered.



**Fig. 2.** The neighbouring system of pixel  $i$  as represented by the filled boxes.

For fixed  $i$ , Eq. (14) admits a simple interpretation within the robust statistic context as being the  $\mathbf{m}_i$  estimator from its neighbourhood using a Cauchy distribution as the influence function (Rey 1983; Press et al. 1992a). This shows its link to  $K_{\text{robust}}(\mathbf{m}_i)$ , which should be taken as the best estimate among the ones presented here. The use of robust statistics estimators as priors in image reconstruction is not new, especially in the context of edge preserving priors (Brette & Idier 1996; Sauer & Bouman 1993). As a successful example of astronomic image deconvolution, (Mugnier et al. 2004) uses the robust “fair” estimator (Rey 1983) for finite difference gradients in the *MISTRAL* algorithm. These kinds of priors reduce the Gibbs oscillations by preferring locally uniform solutions within the chosen scale in the regularization function except for the edges between smooth regions. The main difference of  $K_{\text{Cauchy}}(\mathbf{m})$  with respect to the functions chosen in those references is the asymptotic behaviour. As explained above, the convergence to the maximum likelihood solution is guaranteed in our case, thus our deconvolution tends to be unbiased at the limit  $M \rightarrow \infty$ . Although not explicitly stated, this behaviour is only obtained with usual edge-preserving regularization functions (Brette & Idier 1996; Sauer & Bouman 1993; Mugnier et al. 2004) which generally depend quadratically on the finite difference gradients, by weighting them with respect to the likelihood with an appropriate regularization parameter.

Several open issues have to be settled for the practical use of  $K_{\text{Cauchy}}(\mathbf{m})$ . First we fix the neighbourhood of a given pixel to be the eight ones spatially closer to it, as shown in Fig. 2. Now we should fix the scale values  $s_i$ . As explained above, it measures the a priori precision needed to describe the finite differences between neighbouring pixels, so its the natural scale is of the order of  $\sqrt{M}$ . Furthermore a dependence on the pixel position is not discarded. However, as the proportionality factor is unknown we have adopted the conservative value  $s_i = 1$  as the best that can be done in the worst case, thus the resulting prior will tend to oversmooth the deconvolved solution. Nevertheless, as we shall see, the resulting images are acceptable due to the good asymptotic properties of our regularization. The most important open issue, from the point of view of the author, is the necessity of a multiplicative regularization parameter. In contrast to other arguments for a regularization method, the one we have followed results in a functional closed form without the need of such a parameter if the conjectures about the image are fulfilled. This property is not unique to the algorithmic information approach we follow. For example, (Molina et al. 1992) introduces some

knowledge of the smoothness of the solution, parametrized by a set of parameters similar to our scales. Like ours, their resulting regularization function is not explicitly weighted by a regularization parameter. Nevertheless, as in the case of Eq. (10), a regularization parameter  $\alpha \lesssim 1$  weighting the model algorithmic information could be justified as an improvement to the obtained upper bound estimate of  $K(\mathbf{m})$  in the case where the adopted suppositions about the image in obtaining the algorithmic complexity bounds are known to be wrong. However the correct value for this parameter cannot be computed easily, if it is possible at all. Consequently we again follow the worst case approach by taking  $\alpha = 1$ .

### 3. Implementation and analysis of errors

#### 3.1. Implementation

Although maximization of the functional 3 is not as challenging as the required computation for other deconvolution methods, it is not a simple task. The main difficulty is that  $\log P(\mathbf{d}, \mathbf{m})$  with  $K(\mathbf{m})$  given by Eq. (14) has several local maxima, as inferred by the non global convexity of  $K_{\text{Cauchy}}(\mathbf{m})$ . To avoid them we aim to find an approximate solution as close to the global maximum as possible. Then it use the initial condition on the Polak-Ribiere conjugate gradient algorithm (Press et al. 1992b) in order to find the global maximum. This maximization method has been chosen because it is fast when the partial derivates of the function to be maximized are known in analytical form, which is our case.

In order to obtain the approximate solution we make the following change of variables

$$\mathbf{m}_i = \exp(\mathbf{x}_i). \quad (15)$$

Subsequently, the likelihood  $\log P(\mathbf{d}|\mathbf{m})$  is maximized using the Polak-Ribiere conjugate gradient algorithm with a weak finishing condition. The resulting solution is similar to an accelerated Lucy-Richardson deconvolution with a positivity constrain. Due to the asymptotic properties of  $\log P(\mathbf{d}, \mathbf{m})$ , we expect it to be close to the global maximum. We experimentally verified that the use of this approximate solution in the minimization of the full functional results in acceptable solutions, as we illustrate in Sect. 3.3.

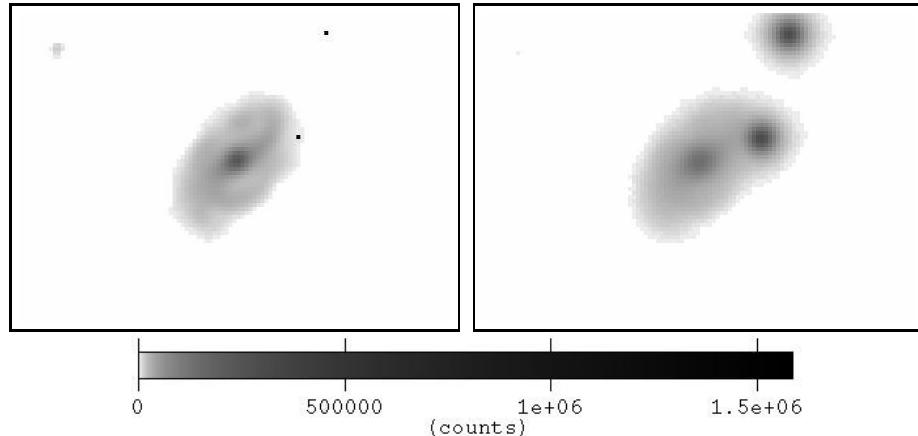
#### 3.2. Error analysis

As the deconvolution process correlates neighbouring pixels, the errors no longer fulfil the usual statistics. Thus their estimate of the errors for the deconvolved image is mandatory in order to exploit it for measurement purposes. A first approach to that estimate is to compute the Hessian matrix of the right hand side of Eq. (3) as an approximation of the correlation matrix, then to diagonalize it to obtain the errors. However the computational cost of that problem is very high. Moreover, it is not guaranteed that this approximation is correct due to the nonlinearity of the problem at hand.

Instead of using the Hessian method, our estimate is inspired by the MINOS approach (see Eadie et al. 1971, for a description): as Eq. (3) admits a probabilistic interpretation, the  $n$  sigma error surface can be defined as the set of models that fulfil the following relation:

$$\log P(\mathbf{d})_{n\sigma} = \log P(\mathbf{d})_{\max} - \frac{n^2}{2} \quad (16)$$

where  $P(\mathbf{d})_{\max}$  is given by Eq. (3) evaluated in its maximum. Once that surface has been identified, the error interval for a



**Fig. 3.** (Left) a synthetic test image with a background galaxy and two stars. (Right) the same image after convolving and adding background Gaussian noise to it. Both figures are in logarithmic scale with a total number of simulated detected photons of  $5 \times 10^6$ .

given pixel can be estimated as the range of values it takes on that surface.

In order to obtain the necessary intervals for this method, we use a Monte-Carlo approach: given the maximum position, i.e. the deconvolved image, we choose a large set of random lines through it, and the intersection of these lines with the 1 sigma surface are computed. With these these values, the error interval for each pixel is easily obtained.

### 3.3. Tests

In order to illustrate the performance of the method, we performed a set of tests. The first ones were on artificial images in order to show how the method deals with the case of very bright sources over smooth backgrounds and how morphological features are recovered. Finally the method was applied to real images just to show the quality of the deconvolution solution in realistic conditions.

#### 3.3.1. Synthetic images

##### Bright sources test

As explained in Sect. 2.2.2, the proposed regularization has been chosen assuming that the images to be deconvolved are piecewise smooth. In the case of astronomical images, this assumption is usually true except for the presence of bright point-like sources. To show that our regularization scheme performs well in this case, we built a set of synthetic images obtained by superposing artificial stars over the image of a true galaxy. The result was subsequently convolved with a PSF of 3.2 pixels FWHM, then sampled using Poisson statistics, and finally Gaussian noise with four photons of rms was added to it. The total number of sampled photons was fixed to  $5 \times 10^6$ . The initial and final  $100 \times 75$  pixels<sup>2</sup> image after this process in the case of adding two artificial stars are shown in Fig. 3.

Top panels in Fig. 4 show the result of the deconvolution of the image on the right panel of Fig. 3 using an implementation of the RL, MEM, and our method, whereas the bottom panels show the deconvolution residuals. With RL and MEM methods, the presence of the bright stars produces the appearance of Gibbs oscillations. In contrast, our method highly attenuates these oscillations recovering the details of the background galaxy and the bright stars. As explained in Sect. 2.2.2, this satisfactory result is due to the fact that our regularization chooses locally smooth solutions, except in the neighbourhood of the pixels where stars have been placed due to its similarity to an edge

between smooth regions. The residuals of the deconvolution with the three methods measures the bias introduced by them. It is clearly seen that our method has a smaller bias around the stars. However, as a consequence of the selection of the worst case scale in Sect. 2.2.2, it oversmooths the center of the galaxy, resulting in a slightly larger bias in that region as compared with the other two methods' residuals.

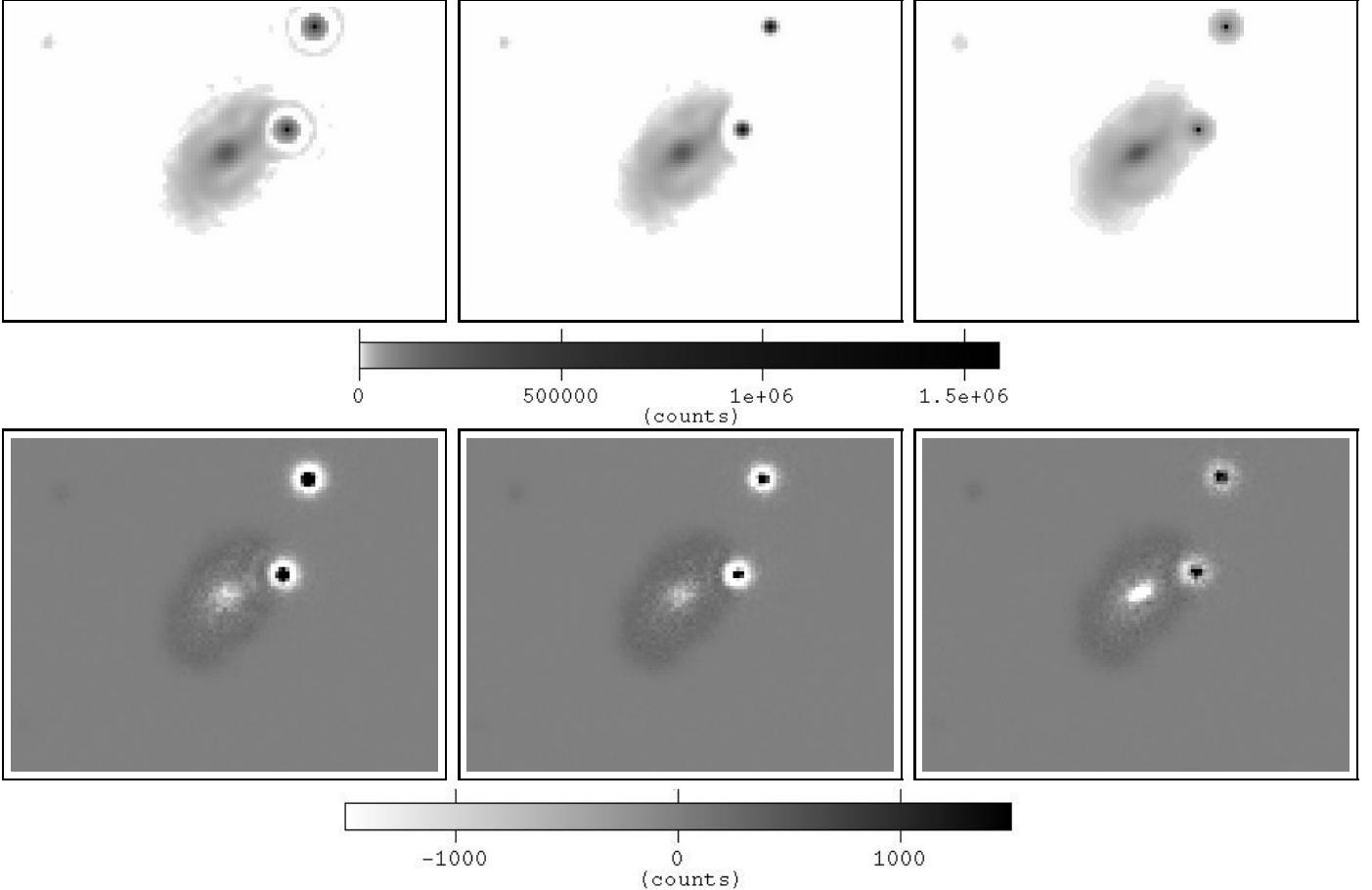
To check the deconvolution in different signal-to-noise ratio (SNR) conditions, several images with a different number of stars were generated. The position of the stars were chosen randomly. Their fluxes were computed as the galaxy total flux multiplied by a uniform random number between zero and one. As the number of sampled photons in each image is kept fixed to  $5 \times 10^6$ , the SNR in the galaxy image decreases as the number of stars increases. In order to quantify the SNR, we adopted the following definition:

$$SNR = \frac{\langle n_{el} \rangle}{\sqrt{\langle n_{el} \rangle + \sigma_{noise}^2}} \quad (17)$$

where  $\langle n_{el} \rangle$  is the mean number of collected electrons per pixel in the galaxy image excluding the stars, and  $\sigma_{noise}$  is the rms of the added Gaussian noise. On the other hand, to measure the quality of the deconvolution, we used the following goodness of fit between the original image and the deconvolved one:

$$G = \frac{1}{N_{pixels}} \sum_i \frac{(t_i - m_i)^2}{t_i + m_i} \quad (18)$$

where  $t$  is the original image before the deconvolution,  $m$  the deconvolved one, and  $N_{pixels}$  is the number of pixels in the image. This quantity has been computed for several regions in the image in order to compare the three methods. The results are shown in Table 1, where the selected regions are the surrounding of the stars (Stars), the galaxy excluding those surroundings (Galaxy) and the image background (Background). Clearly our solution is the best among the three for the Stars and the Background regions. This is because it is specially well-suited to describing uniform regions, like the Background, and the bias due to the regularization is small for edges between smooth regions, like the Stars. Furthermore, in spite of the oversmoothing due to the worst case scale chosen in Sect. 2.2.2, the Galaxy region deconvolution with our method is as good as the MEM one. Finally there is an apparent contradictory behaviour for the 20 stars case, that must be clarified: the deconvolution of the



**Fig. 4.** Top panels: deconvolution of the image on the right of Fig. 3 using the MEM method (left), the RL one (center), and ours (right). Bottom panels: residual of the deconvolution, defined as  $d_i - \sum_{j=1}^N R_i^j m_j$  using the notation of Sect. 2, for the corresponding solutions on the top panels.

Background and the Stars regions for our method shows a large improvement as compared with the results for less stars. This is because the galaxy is dim enough for 20 stars to be statistically compatible with an almost uniform region. In consequence, the image properties are especially close to the assumptions in determining our regularization function. Furthermore, the worst-case scale parameter is very well-suited to deconvolving uniform regions, which is the case for 20 stars.

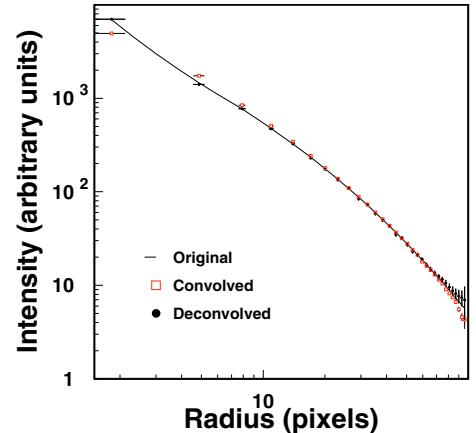
### 3.3.2. Radial profile test

In this test we simulated the observation of 500 elliptical galaxies with a single telescope. As its PSF we used the same as the previous test, as well as the sky background properties. The observation is such that the number of collected electrons per image is around  $10^5$ . As the galaxy brightness radial profiles, we used the Sérsic model (Sérsic 1968):

$$I(r) = I(0) \exp\left(-b(n)\left(\frac{r}{r_e}\right)^{\frac{1}{n}}\right) \quad (19)$$

where  $I(0)$  is the brightness at  $r = 0$ ,  $r_e$  a radial scale,  $n$  the Sérsic index, and  $b(n)$  is a function that can be approximated as  $b(n) \approx 2n - 0.324$  for  $n > 1$  (Trujillo et al. 2001). In our simulations these parameters range from 0 to 60 pixels for  $r_e$  and from 2.5 to 5.5 for the Sérsic index  $n$ , whereas the resulting image has a size of  $152 \times 120$  pixels.

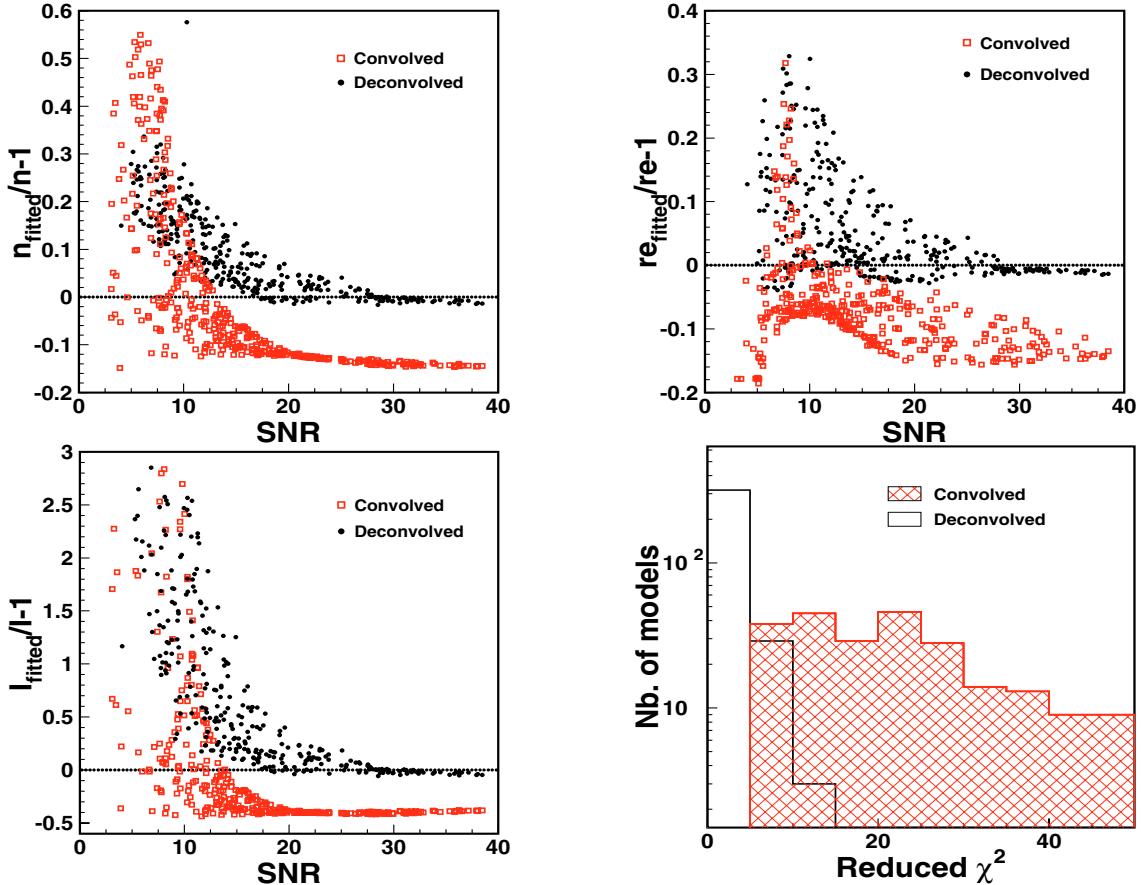
Figure 5 shows the simulated profile, the one obtained from the convolved image and the result of the deconvolution for a



**Fig. 5.** Profile for a simulated image with parameters  $n = 3.6$  and  $r_e = 27$  pixels (line), the convolved one (squares), and the deconvolved (dots). The errors shown for the profile obtained from the deconvolved image have been obtained from quadrature addition of the errors estimate explained in Sect. 3.2.

sample case. Clearly the algorithm correctly recovers the intensity close to the galaxy center and on its edge.

To quantify the improvement due to the method, we fitted the light profiles before and after the deconvolution to the Sérsic model using the  $\chi^2$  method. The obtained parameters were compared with the simulated ones as a function of the SNR defined in Eq. (17). The results are shown in Fig. 6. As expected, the deconvolved radial profile gets closer to the true one as the SNR



**Fig. 6.** Result of the fit to the Sérsic model given by Eq. (19) of the convolved (squares) and deconvolved (dots) image radial profiles, as explained in the text. The top left panel shows the relative deviation of the obtained value for the  $n$  parameter compared to the true one. The relative deviation for the parameters  $r_e$  and  $i$  are plotted in the top right and bottom left panels, respectively. Finally the bottom right panel shows the obtained reduced  $\chi^2$  distribution for the fits.

**Table 1.** Goodness of fit of the deconvolution using synthetic images computed for the three methods and different regions in the image as explained in the text. The signal-to-noise ratio of the galaxy image is shown in the second column, whereas the number of artificial stars appears in the first.

Stars	Galaxy SNR	Stars			Galaxy			Background		
		MEM	RL	Focus	MEM	RL	Focus	MEM	RL	Focus
0	16.1	—	—	—	44	9.4	49	1.4	2.0	1.3
2	11.1	$1.6 \times 10^5$	$1.9 \times 10^5$	$2.0 \times 10^4$	17	9.8	17	60	82	20
10	5.8	$7.4 \times 10^4$	$5.4 \times 10^4$	$3.6 \times 10^4$	11	5.2	15	100	100	64
20	3.9	$3.4 \times 10^4$	$3.7 \times 10^4$	$2.7 \times 10^2$	8.3	2.6	6.5	96	120	9.2

increases for all the parameters of the Sérsic model. In contrast, the radial profile for the convolved image has a large bias for all the parameters independent of the SNR value. Furthermore, the  $\chi^2$  distribution (Fig. 6, bottom right panel) suggests that the Sérsic model is no longer a good description of the light profile for the convolved image, as opposed to the deconvolved image.

### 3.3.3. Real images

Previous tests with synthetic images have shown that the proposed method correctly recovers the original image properties; however, level of detail and complexity of synthetic images is far from the level in real images. To further test the performance of the method we deconvolved two images of real galaxies to show how the deconvolution method unveils hidden structures due to the PSF of the telescope.

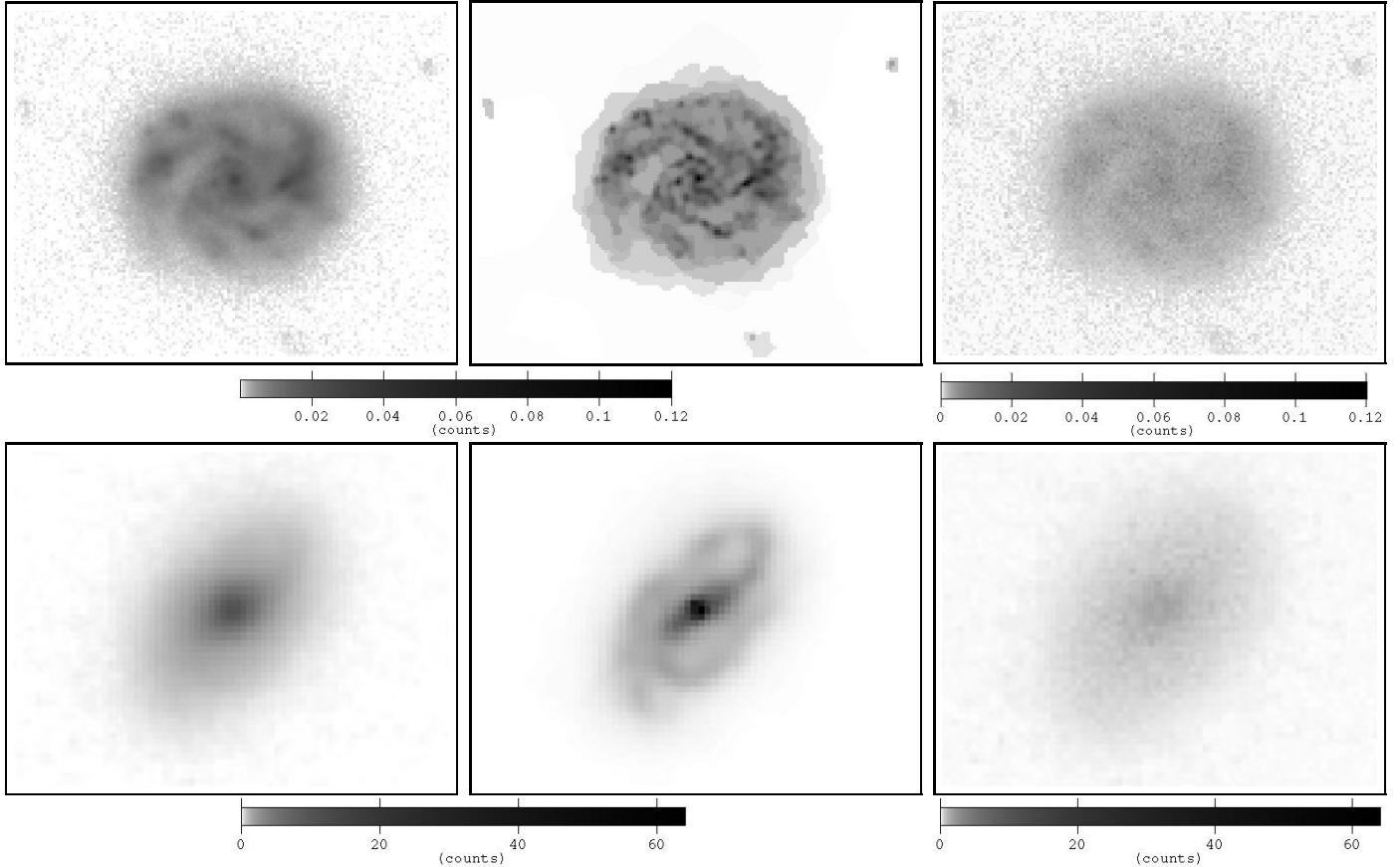
The top panels of Fig. 7 show the original, deconvolved and estimated errors for the galaxy catalogued with id. 33791

on the COMBO-17 CDF-S catalog (Trujillo & Pohlen 2005; Wolf et al. 2004). The  $0.34'' \times 0.46''$  original image, obtained with the Advance Camera for Surveys of the Hubble Space Telescope, was extracted from the publicly available  $z$ -band mosaics. Clearly the level of detail of the deconvolved image is higher than the original one. Furthermore, the deconvolution process does not introduce artifacts on the sky background.

The bottom panels of Fig. 7 show the deconvolution method carried out on a processed  $1'' \times 5''$   $H$ -band image obtained with the ISAAC instrument on the ESO-VLT-UT1 telescope. In this case the spiral galaxy structure, that was hardly visible in the original image, was revealed after the method was applied.

## 4. Conclusions

We have proposed a deconvolution method here based on searching for the simplest solution that is statistically compatible with the data. In so doing, we have defined what we mean by “simplest” in terms of the Kolmogorov algorithmic complexity. Our



**Fig. 7.** Top panels: (left)  $z$ -band image of a test galaxy obtained with the Wide Field Camera of the Hubble Space Telescope (see text for details). (Center) deconvolved image of the top left galaxy. (Right) pixel-by-pixel estimated error for the top center deconvolved image. Bottom panels: (Left)  $H$ -band image of a test galaxy obtained with the ISAAC instrument at the ESO-VLT-UT1 telescope. (Center) deconvolved image of the bottom left galaxy. (Right) pixel-by-pixel estimated error for the bottom center deconvolved image. In all cases the counts are given in arbitrary units.

tests suggest that our deconvolution correctly recovers the structures on the original image that have been lost in the convolution process. We also observed that our method is robust against the introduction of artifacts in the deconvolved image.

The implementation of a pixel-by-pixel error analysis that correctly takes the correlations into account, allows us to use the deconvolved images to measure quantities that involve a set of pixels with a correct error estimation.

An implementation of the algorithm can be requested by email to the author or at the URL <http://focus-algorithm.sourceforge.net/>.

**Acknowledgements.** We want to thank Ignacio Trujillo for all the interesting discussions about the deconvolution issue, his comments on the paper, and for kindly providing the real galaxy images we used. These are based on observations made with ESO Telescopes at the La Silla or Paranal Observatories under program ID 164.O-0612(A) and on observations made with the NASA/ESA Hubble Space Telescope, obtained from the data archive at the Space Telescope Institute. STScI is operated by the association of Universities for Research in Astronomy, Inc. under the NASA contract NAS 5-26555.

## References

- Ables, J. G. 1974, *A&AS*, 15, 383
- Brette, S., & Idier, J. 1996, in Proceedings of the International Conference on Image Processing, Lausanne, Switzerland, 65
- Eadie, W. T., Drijar, D., James, F., Roos, M., & Sadoulet, B. 1971, *Stat. Meth. Exper. Phys.* (North-Holland)
- Gull, S. F., & Skilling, F. 1991, MEMSYS5 Quantized Maximum Entropy User's Manual
- Kolmogorov, A. N. 1965, *Probl. Inf. Transm.*, 1, 1
- Kullback, S., & Leibler, R. A. 1951, *Ann. Math. Stat.*, 22(1), 79
- Levin, L. A. 1974, *Probl. Inf. Transm.*, 10, 206
- Li, M., & Vitanyi, P. 1997, *An Introduction to Kolmogorov Complexity and Its Applications* (Springer Verlag)
- Lucy, R. B. 1974, *AJ*, 79, 745
- Magain, P., Courbin, F., & Sohy, S. 1998, *ApJ*, 494, 472
- Molina, R., Ripley, B. D., Molina, A., Moreno, F., & Ortiz, J. L. 1992, *AJ*, 104, 1662
- Mugnier, L. M., Fusco, T., & Conan, J. 2004, *J. Opt. Soc. Am. A*, 21, 1841
- Press, W. H., Flannery, B. P., Teukolsky, S. A., & Vetterling, W. T. 1992a, *Numerical Recipes: The Art of Scientific Computing*, 2nd edn. (Cambridge University Press)
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1992b, *Numerical Recipes in C: The Art of Scientific Computing* (New York, NY, USA: Cambridge University Press)
- Puetter, R. C., & Yahil, A. 1999, in *Astronomical Data Analysis Software and Systems VIII*, ed. D. M. Mehringer, R. L. Plante, & D. A. Roberts, *ASP Conf. Ser.*, 172, 307
- Reiter, J. 1990, *NASA STI/Recon Technical Report*, 91, 22953
- Rey, W. J. J. 1983, *Introduction to Robust and Quasi-Robust Statistical Methods* (Berlin: Springer-Verlag)
- Richardson, W. H. 1972, *J. Opt. Soc. Am.*, 62, 55
- Risanen, G. 1983, *Ann. Stat.*, 11, 416
- Sauer, K., & Bouman, C. 1993, *A local update strategy for iterative reconstruction from projections*
- Sault, R. J. 1990, *ApJ*, 354, L61
- Sérsic, J. 1968, *Atlas de Galaxias Australes* (Córdoba: Observatorio Astronómico)
- Trujillo, I., & Pohlen, M. 2005, *ApJ*, 630, L17
- Trujillo, I., Graham, A. W., & Caon, N. 2001, *MNRAS*, 326, 869
- Wernecke, S. J. 1977, *Rad. Sci.*, 12, 831
- Wernecke, S. J., & D'Addario, L. R. 1977, *IEEE Trans. Comm.*, 26, 351
- Wolf, C., Meisenheimer, K., Kleinheinrich, M., et al. 2004, *A&A*, 421, 913