Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Journal of Computational and Applied Mathematics 197 (2006) 29–43

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

Constrained iterations for blind deconvolution and convexity issues

Giulia Spaletta[a,*], Luca Caucci[b]

[a] Mathematics Department, Bologna University, Piazza Porta S. Donato 5, 40127 Bologna, Italy
[b] Electrical and Computer Engineering Department, University of Arizona, Tucson, USA

Received 24 August 2005; received in revised form 10 October 2005

Abstract

The need for image restoration arises in many applications of various scientific disciplines, such as medicine and astronomy and, in general, whenever an unknown image must be recovered from blurred and noisy data [M. Bertero, P. Boccacci, Introduction to Inverse Problems in Imaging, Institute of Physics Publishing, Philadelphia, PA, USA, 1998]. The algorithm studied in this work restores the image without the knowledge of the blur, using little a priori information and a *blind inverse filter* iteration. It represents a variation of the methods proposed in Kundur and Hatzinakos [A novel blind deconvolution scheme for image restoration using recursive filtering, IEEE Trans. Signal Process. 46(2) (1998) 375–390] and Ng et al. [Regularization of RIF blind image deconvolution, IEEE Trans. Image Process. 9(6) (2000) 1130–1134]. The problem of interest here is an *inverse* one, that cannot be solved by simple filtering since it is ill-posed. The imaging system is assumed to be linear and space-invariant: this allows a simplified relationship between unknown and observed images, described by a *point spread function* modeling the distortion. The blurring, though, makes the restoration ill-conditioned: *regularization* is therefore also needed, obtained by adding constraints to the formulation. The restoration is modeled as a constrained minimization: particular attention is given here to the analysis of the objective function and on establishing whether or not it is a convex function, whose minima can be located by classic optimization techniques and descent methods. Numerical examples are applied to simulated data and to real data derived from various applications. Comparison with the behavior of methods [D. Kundur, D. Hatzinakos, A novel blind deconvolution scheme for image restoration using recursive filtering, IEEE Trans. Signal Process. 46(2) (1998) 375–390] and [M. Ng, R.J. Plemmons, S. Qiao, Regularization of RIF Blind Image Deconvolution, IEEE Trans. Image Process. 9(6) (2000) 1130–1134] show the effectiveness of our variant.

MSC: 65R32; 65T50; 92C55

Keywords: Blind restoration; Inverse problems; Regularization; Constrained minimization; Fourier transform

1. Introduction

Blind restoration is the process of estimating both the original image and the blur from the observed picture, using partial information on the imaging system. Assuming linearity and space-invariance of the image formation system, the distorted noise-free image can be expressed as the convolution of the true image with the blurring function. If the blur is known, the *Fourier Transform* (FT) can be used to invert the degradation process and compute the deconvolution that will yield the original image.

[*] Corresponding author.
E-mail address: spaletta@dm.unibo.it (G. Spaletta).

0377-0427/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2005.10.020

The blind restoration method analyzed here aims to recover the image without the knowledge of the blur, using a priori constraints and an inverse filter. The algorithm is called *Non-negative*, *Support and PSF constrained Recursive Inverse Filtering* and represents a variation of the methods proposed in [9] (derived from the Master's thesis [8]) and [10]. A prototype software has been realized [12] to test our algorithm and the methods in [9,10] both on simulated and real data.

This paper is organized as follows. In Section 2 the restoration problem is outlined and the difficulties that arise in solving it are summarized. Section 3 contains a description of our algorithm, that will be denoted by the acronym NSH–RIF. Convexity issues are considered in Section 4. After restoration, a *Total Variation regularization* (TV) technique can be used to sharpen the image, as recalled in Section 5. Related to the case of astronomic imaging, a procedure to obtain an initial approximation for the unknown blurring function is given in Section 6. Section 7 collects numerical examples that show the effectiveness of our algorithm.

2. Background

Let $f(x)$ be the original image to be reconstructed. For simplicity, we consider first the one-dimensional (1D) case, but all concepts can be extended to the two-dimensional (2D) case [4]. The observed image $\tilde{g}(x)$ can be written as

$$\tilde{g}(x) = \int_{\mathbb{R}} \hbar(\zeta, x, f(x))\, \mathrm{d}\zeta + \delta(x), \tag{1}$$

where $\hbar(\zeta, x, f)$ is the *point spread* function (PSF), modeling the blur in a general nonlinear and space-dependent imaging situation, and $\delta(x)$ is the randomly distributed noise. If the image formation system is linear and space-invariant—a condition that can be assumed to hold in most practical cases—(1) simplifies to

$$\begin{aligned}\tilde{g}(x) &= (h \circ f)(x) + \delta(x)\\ &= g(x) + \delta(x),\end{aligned} \tag{2}$$

where $\circ$ denotes the convolution operator, $h(x)$ is the PSF representing the linear and space invariant blur, and $g(x)$ is the blurred noise-free image.

In a discretized imaging problem, let a real n-dimensional vector $\mathfrak{f}$ store the positive values (i.e., the pixel intensities) of $f(x)$, while $\mathfrak{h}$ is a real vector whose $2m+1$ components represent the values of the unknown function $h(x)$. The discrete analog of the convolution operator (2) can be efficiently computed using FTs (see Definition 2.1). In order to avoid overlapping in the direct and inverse transforms, it is necessary to *extend* both $\mathfrak{f}$ and $\mathfrak{h}$ at least to the dimension $2m+n$:

$$\boldsymbol{f} = \begin{pmatrix} 0_{m\times n} \\ I_n \\ 0_{m\times n} \end{pmatrix} \mathfrak{f}, \quad \boldsymbol{h} = \begin{pmatrix} 0_{(m+1)\times m} & I_{m+1} \\ 0_{(n-1)\times m} & 0_{(n-1)\times(m+1)} \\ I_m & 0_{m\times(m+1)} \end{pmatrix} \mathfrak{h}. \tag{3}$$

The integers n and m are, respectively, called the *support* of $\boldsymbol{f}$ and $\boldsymbol{h}$, which now have dimensions $2m+n$. The discretization of $\tilde{g}(x)$, $g(x)$ and $\delta(x)$ in (2) are represented by real vectors $\tilde{\boldsymbol{g}}$, $\boldsymbol{g}$ and $\boldsymbol{\delta}$, all of dimension $2m+n$.

Definition 2.1. The direct and inverse discrete Fourier transforms $\mathscr{F}$ and $\mathscr{F}^{-1}$ are defined as

$$\boldsymbol{v} = \mathscr{F}(\boldsymbol{u}), \quad v_j = (n)^{-1/2} \sum_{k=1}^{n} u_k \mathrm{e}^{2\pi \mathrm{i}(k-1)(j-1)/n},$$

$$\boldsymbol{u} = \mathscr{F}^{-1}(\boldsymbol{v}), \quad u_k = (n)^{-1/2} \sum_{j=1}^{n} v_j \mathrm{e}^{-2\pi \mathrm{i}(k-1)(j-1)/n},$$

where $\boldsymbol{v} = (v_1, \ldots, v_j, \ldots, v_n)^{\mathrm{T}}$ and $\boldsymbol{u} = (u_1, \ldots, u_k, \ldots, u_n)^{\mathrm{T}}$ are complex vectors.

The discrete convolution $\boldsymbol{g} = \boldsymbol{h} \circ \boldsymbol{f}$ can be expressed as

$$\boldsymbol{g} = \sqrt{2m+n}\mathscr{F}^{-1}(\mathscr{F}(\boldsymbol{h}) \times \mathscr{F}(\boldsymbol{f})),$$

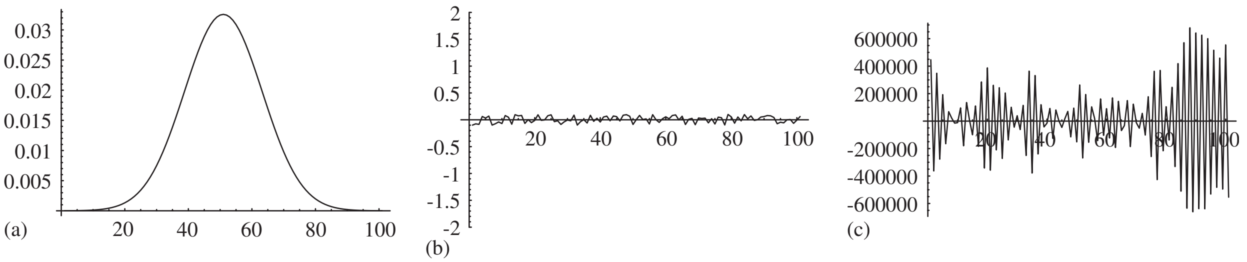

Fig. 1. Example of noise amplification: (a) Function $\boldsymbol{h}$; (b) Noise $\boldsymbol{\delta}$; (c) Error $\tilde{\boldsymbol{f}} - \boldsymbol{f}$.

where the symbol × denotes the componentwise multiplication of two vectors. In matrix form

$$\boldsymbol{g} = H\boldsymbol{f} = WDW^{-1}\boldsymbol{f},$$

where H is the circulant matrix of the entries of $\boldsymbol{h}$, D is the diagonal matrix of the direct FT of the first column of H, while W is the symmetric matrix of the eigenvectors of H. It can be shown that the application of W to a vector $\boldsymbol{w}$ is equivalent to computing $\mathscr{F}^{-1}(\boldsymbol{w})$, while $W^{-1}\boldsymbol{w}$ yields $\mathscr{F}(\boldsymbol{w})$.

Given $\boldsymbol{g}$ and $\boldsymbol{h}$, the original image $\boldsymbol{f}$ can then be recovered

$$\boldsymbol{f} = (2m+n)^{-1/2}\mathscr{F}^{-1}(\mathscr{F}(\boldsymbol{g}) \div \mathscr{F}(\boldsymbol{h})), \tag{4}$$

where the symbol ÷ denotes the componentwise division of two vectors.

Unfortunately (4) cannot be easily exploited. Image restoration is a hard problem, belonging to the class of the so called *inverse problems*, that may not have a (unique) solution or may possess a solution that is very sensitive to perturbation of the input data [1,3]. Noise can prevent the existence of the solution, since it may modify the observed image so that it can no longer be expressed as a convolution: noise undermines the relationship among $\boldsymbol{f}$, $\boldsymbol{g}$ and $\boldsymbol{h}$, making an exact restoration impossible.

In practical situations $\boldsymbol{\delta}$ is not known and so we are forced to consider $\tilde{\boldsymbol{g}}$ instead of $\boldsymbol{g}$ in (4). By the linearity property of the Fourier transform [4]

$$\begin{aligned}\tilde{\boldsymbol{f}} &= (2m+n)^{-1/2}\mathscr{F}^{-1}(\mathscr{F}(\tilde{\boldsymbol{g}}) \div \mathscr{F}(\boldsymbol{h}))\\ &= (2m+n)^{-1/2}\mathscr{F}^{-1}(\mathscr{F}(\boldsymbol{\delta}+\boldsymbol{g}) \div \mathscr{F}(\boldsymbol{h}))\\ &= (2m+n)^{-1/2}\mathscr{F}^{-1}(\mathscr{F}(\boldsymbol{\delta}) \div \mathscr{F}(\boldsymbol{h})) + \boldsymbol{f},\end{aligned}$$

where $\tilde{\boldsymbol{f}} \in \mathbb{R}^{2m+n}$ is the image that has been restored from $\tilde{\boldsymbol{g}}$.

If some of the entries of $\mathscr{F}(\boldsymbol{h})$ are much smaller in magnitude than the corresponding entries in $\mathscr{F}(\boldsymbol{\delta})$, then some components of the error $\tilde{\boldsymbol{f}} - \boldsymbol{f}$ can have very large magnitude, leading to noise amplification in the restored image. Fig. 1 shows the amplified error that may occur, when applying the deconvolution described in Section 3 to the noise component of a signal. *Regularization* [1,2] is used to control this amplification. Constraints are then imposed which follow physical requirements, such as the observation that both image and PSF values should be positive, as well as restrictions on the integer supports n and m. The fact that the PSF is band-limited is employed in a thresholding procedure, by giving low weight or even ignoring those components of $\boldsymbol{h}$ from which noise amplification may stem.

The *Conjugate Gradient iteration* can be viewed as a restoration method, because low frequency image components converge much faster than high frequency components.

The drawback in suppressing high frequency information lies in the consequent loss of image contrast and resolution. A post-processing procedure, described in Section 5, is applied to improve the quality of the reconstructed data.

3. Blind deconvolution algorithm

'*Blind*' *deconvolution* refers to recovering both the original image and the PSF from the observed image, trying to avoid noise amplification. This is harder than the so-called *classic deconvolution* (in which $\boldsymbol{h}$ is known) and generally requires the introduction of a priori information. Fundamental to our NSH–RIF algorithm is the knowledge of the support of the true image; when the exact support is not known at the initial iteration, it is estimated from the observed image and used as an input.

The algorithm seeks a solution that satisfies:

- non-negative and image support constraints;
- noise amplification constraints;
- energy conservation constraints;
- PSF support constraints.

By enforcing these constraints an *inverse filter* is computed, namely $\mathfrak{s} \in \mathbb{R}^{2p+1}$, with $2p+1 \leqslant 2m+n$. The vector $\mathfrak{s}$ is called 'inverse filter' because its extension and rearrangement $\boldsymbol{s} \in \mathbb{R}^{2m+n}$ can be interpreted as the inverse of the PSF with respect to the convolution operator

$$\boldsymbol{s} = (2m+n)^{-1}\mathscr{F}^{-1}(\boldsymbol{e} \div \mathscr{F}(\boldsymbol{h})), \tag{5}$$

where $\boldsymbol{e} = (1, 1, \ldots, 1)^{\mathrm{T}} \in \mathbb{R}^{2m+n}$. The integer p is called the *support* of $\boldsymbol{s}$. Setting $q = (2m+n) - 2p$, the extension from $\mathfrak{s}$ to $\boldsymbol{s}$ is performed as follows:

$$\boldsymbol{s} = Z\mathfrak{s}, \quad Z \overset{\text{def}}{=} \begin{pmatrix} 0_{(p+1)\times p} & I_{p+1} \\ 0_{(q-1)\times p} & 0_{(q-1)\times(p+1)} \\ I_p & 0_{p\times(p+1)} \end{pmatrix}. \tag{6}$$

Given $\tilde{\boldsymbol{g}}$ and computed $\mathfrak{s}$, the estimated image $\tilde{\boldsymbol{f}}$ can be expressed as a convolution and efficiently recovered

$$\tilde{\boldsymbol{f}} = \sqrt{2m+n}\mathscr{F}^{-1}(\mathscr{F}(\tilde{\boldsymbol{g}}) \times \mathscr{F}(\boldsymbol{s})). \tag{7}$$

In further iterations, both PSF and image estimate are updated together.

To take into account all constraints, a non-negative cost function is constructed

$$J_{\tilde{\boldsymbol{g}},\boldsymbol{\mu}}(\mathfrak{s}) = \phi_{\tilde{\boldsymbol{g}}}(\mathfrak{s}) + \psi_{\mu_1,\mu_2}(\mathfrak{s}) + \chi_{\mu_3}(\mathfrak{s}) + \omega_{\mu_4}(\mathfrak{s}), \tag{8}$$

where $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3, \mu_4)^{\mathrm{T}}$ collects four real non-negative parameters, that will be used to tune the constraints: they are effectively *regularization* parameters.

The value of $J_{\tilde{\boldsymbol{g}},\boldsymbol{\mu}}(\mathfrak{s})$ measures the quality of the inverse filter $\mathfrak{s}$. Locating a small value of the cost function means that the corresponding restored image satisfies the constraints and represents an acceptable approximation of the true image, in a way that becomes more and more stringent with the decrease of $J_{\tilde{\boldsymbol{g}},\boldsymbol{\mu}}(\mathfrak{s})$.

Each term in (8) will now be described.

3.1. Non-negative and image support constraint

The first term $\phi_{\tilde{\boldsymbol{g}}}(\mathfrak{s})$ corresponds to non-negative and support constraints; these have been proposed in [9] and used also in [10]. Let n be the estimated support of the restored image, that is to say that only the central n elements of $\tilde{\boldsymbol{f}} = (\tilde{f}_1, \ldots, \tilde{f}_{2m+n})^{\mathrm{T}}$ may be non-zero; furthermore, they should not assume negative values since they represent the intensity of the beams captured by the instruments. Given n and $(2m+n)$ in such a way that m is a positive integer, we can define a square real matrix V of order $(2m+n)$ as

$$V = (v_{k,j}), \quad v_{k,j} = \begin{cases} 0 & \text{if } k = j \text{ and } m < k \leqslant (m+n) \text{ and } \tilde{f}_k > 0, \\ 1 & \text{otherwise,} \end{cases}$$

and define a vector of dimension $2m+n$ as

$$z_{\text{supp}} = V\tilde{\boldsymbol{f}}.$$

Recalling (6) and (7), the first term in the cost function depends on $\mathfrak{s}$ and is given by

$$\phi_{\tilde{\boldsymbol{g}}}(\mathfrak{s}) = \|z_{\text{supp}}\|_2^2, \tag{9}$$

which is a continuous, convex, but not everywhere differentiable function.

3.2. *Noise amplification constraint*

In Section 2 we saw that the small magnitude components of $\mathscr{F}(\boldsymbol{h})$ can lead to noise amplification. From (5) it follows that

$$\mathscr{F}(\boldsymbol{s}) = (2m+n)^{-1}(\boldsymbol{e} \div \mathscr{F}(\boldsymbol{h})).$$

In order to penalize those elements of $\mathscr{F}(\boldsymbol{h})$ of small magnitude, we must penalize the elements of $\mathscr{F}(\boldsymbol{s})$ whose magnitudes are larger than a given threshold $\mu_1 \geqslant 0$. This can be accomplished by introducing the matrix

$$W = (w_{k,j}), \quad w_{k,j} = \begin{cases} 0 & \text{if } k = j \text{ and } |(\mathscr{F}(\boldsymbol{s}))_k| < \mu_1, \\ 1 & \text{otherwise,} \end{cases}$$

and constructing a $(2m+n)$-dimensional vector as

$$z_{\text{reg}} = W\mathscr{F}(\boldsymbol{s}).$$

Recalling (6), the second term in the cost function can be defined as

$$\psi_{\mu_1,\mu_2}(\mathfrak{s}) = \mu_2\|z_{\text{reg}}\|_2^2, \tag{10}$$

where $\mu_2 \geqslant 0$. The two real parameters μ_1 and μ_2 are estimated heuristically and used to adjust the control on the noise amplification.

Constraints (10) has been proposed in [10] as convex ones but, as shown in Section 4, the function $\psi_{\mu_1,\mu_2}(\mathfrak{s})$ is *quasi-convex* (see Definition 4.1) and has a number of discontinuities of the order of the size of $\boldsymbol{s}$.

3.3. *Energy conservation constraint*

The term $\chi_{\mu_3}(\mathfrak{s})$ forces the inverse filter to be *conservative*, that is to say the energy of the restored image must be equal to the energy of the observed image. At the same time it avoids the trivial solution $\mathfrak{s} = \mathbf{0}_{2p+1}$. To achieve these goals, $\chi_{\mu_3}(\mathfrak{s})$ is defined as

$$\chi_{\mu_3}(\mathfrak{s}) = \mu_3\left(\sum_{k=1}^{2p+1} \mathfrak{s}_k - 1\right)^2, \tag{11}$$

where the real parameter $\mu_3 \geqslant 0$ has to be estimated heuristically. The function $\chi_{\mu_3}(\mathfrak{s})$ is infinitely differentiable and convex. A similar approach has been proposed in [9,10].

3.4. *PSF support constraint*

The last term $\omega_{\mu_4}(\mathfrak{s})$ imposes constraints on the support of the extended PSF, which is estimated from (5) as follows:

$$\boldsymbol{h} = (2m+n)^{-1}\mathscr{F}^{-1}(\boldsymbol{e} \div \mathscr{F}(\boldsymbol{s})).$$

Only the first $(m+1)$ and the last m elements of $\boldsymbol{h}$ may be non-zero. By constructing

$$U=(u_{k,j}),\quad u_{k,j}=\begin{cases}0 & \text{if } k=j \text{ and } (m+1)<k<(m+n),\\ 1 & \text{otherwise,}\end{cases}$$

and building a vector with $2m+n$ components as

$$z_{\mathrm{PSF}}=U\boldsymbol{h},$$

we can force the estimated $\boldsymbol{h}$ to converge to the desired form, during the minimization, by defining

$$\omega_{\mu_4}(\mathfrak{s})=\mu_4\|z_{\mathrm{PSF}}\|_2^2. \tag{12}$$

The real parameter $\mu_4 \geqslant 0$ is used here to tune the PSF support constraint. The function $\omega_{\mu_4}(\mathfrak{s})$ is convex and continuous.

These PSF support constraints were introduced to try and force additional convexity to the cost function. Their effectiveness in improving the convergence properties of our algorithm, when compared with the methods in [9,10], is shown by the numerical experiments of Section 7.

4. Convexity analysis

Certain choices of the parameters μ_1, μ_2, μ_3 and μ_4 yield restoration methods that already have been discussed in the literature. For instance, setting $\mu_2=\mu_4=0$ is equivalent to considering the NAS–RIF technique described in [9], while choosing $\mu_4=0$ yields the NSR–RIF method proposed in [10]. The conditions under which these methods are globally convergent are related to the Hessian of the cost function being positive definite.

The use of the noise amplification constraint (10), employed by our method and NSR–RIF, may cause the cost function $J_{\tilde{\boldsymbol{g}},\boldsymbol{\mu}}(\mathfrak{s})$ to be only locally convex. This can be seen by considering a simplified case in which the inverse filter is a scalar $\mathfrak{s}\in\mathbb{R}$. In this case, $J_{\tilde{\boldsymbol{g}},\boldsymbol{\mu}}\colon \mathbb{R}\to\mathbb{R}^+$ and also $\psi_{\mu_1,\mu_2}\colon \mathbb{R}\to\mathbb{R}^+$. A plot of $\psi_{\mu_1,\mu_2}(\mathfrak{s})$ for $\mu_1=2$ and $\mu_2=1$, is given in Fig. 2: there exist two points of discontinuity and $\psi_{\mu_1,\mu_2}(\mathfrak{s})$ is quasi-convex [6].

Definition 4.1. A function $\psi\colon \mathbb{R}^k\to\mathbb{R}$ is quasi-convex if for all $\boldsymbol{v},\boldsymbol{w}\in\mathbb{R}^k$ and $t\in[0,1]$

$$\psi(t\boldsymbol{v}+(1-t)\boldsymbol{w})\leqslant \max\{\psi(\boldsymbol{v}),\psi(\boldsymbol{w})\}.$$

Now consider the case of $\mathfrak{s}\in\mathbb{R}^2$, $\psi_{\mu_1,\mu_2}\colon\mathbb{R}^2\to\mathbb{R}^+$, with $\mu_1=3$ and $\mu_2=1$. Fig. 3(a) shows a plot on the domain $[-6,6]\times[-6,6]$. For fixed $\boldsymbol{v},\boldsymbol{w}\in\mathbb{R}^2$ and $t\in[0,1]$, the expression $\psi_{\mu_1,\mu_2}(t\,\boldsymbol{v}+(1-t)\,\boldsymbol{w})$ represents the intersection of the surface $\psi_{\mu_1,\mu_2}(\mathfrak{s})$ with the plane containing the points $(v_1,v_2,0)^{\mathrm{T}}$, $(w_1,w_2,0)^{\mathrm{T}}$ and parallel to the vertical axis, as illustrated in Fig. 3(b). The result of this operation is a *piecewise* convex (not quasi-convex) function, depicted in Fig. 3(c). Depending on the choice of μ_1 and μ_2, therefore, $\psi_{\mu_1,\mu_2}(\mathfrak{s})$ may be a non-convex (not even quasi-convex) function.

This behavior is inherited by the cost function $J_{\tilde{\boldsymbol{g}},\boldsymbol{\mu}}(\mathfrak{s})$. We are then faced with a choice of which minimization algorithm to employ. In this work a Conjugate Gradient (CG) iteration is used (see Section 4.1) and the solution is not guaranteed to be a global minimum, but rather a local one.

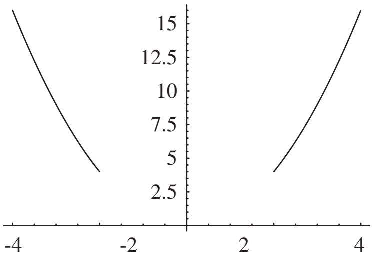

Fig. 2. Non-continuity and quasi-convexity of $\psi_{\mu_1,\mu_2}(\mathfrak{s})$.

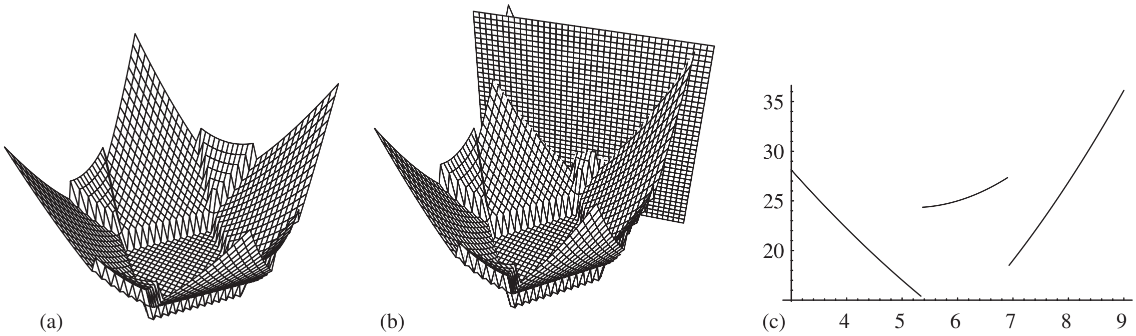

Fig. 3. Piecewise convexity of $\psi_{\mu_1,\mu_2}\colon \mathbb{R}^2 \to \mathbb{R}^+$ with $\mu_1 = 3$ and $\mu_2 = 1$: (a) Plot of ψ_{μ_1,μ_2}; (b) Intersection with a plane; (c) Resulting curve.

4.1. Minimization issues

The Polak–Ribière CG iteration [11] is employed to compute the optimal inverse filter. At each iteration, the gradient $\nabla J_{\tilde{\boldsymbol{g}},\boldsymbol{\mu}}(\mathfrak{s})$ of $J_{\tilde{\boldsymbol{g}},\boldsymbol{\mu}}(\mathfrak{s})$ with respect to $\mathfrak{s}$ must be computed; from (8) it follows:

$$\nabla J_{\tilde{\boldsymbol{g}},\boldsymbol{\mu}}(\mathfrak{s}) = \nabla\phi_{\tilde{\boldsymbol{g}}}(\mathfrak{s}) + \nabla\psi_{\mu_1,\mu_2}(\mathfrak{s}) + \nabla\chi_{\mu_3}(\mathfrak{s}) + \nabla\omega_{\mu_4}(\mathfrak{s}).$$

Letting

$$P = (I_{2p+1} \quad 0_{(2p+1)\times(q-1)}), \qquad M = \begin{pmatrix} 0_{m\times(m+n)} & I_m \\ I_{m+n} & 0_{(m+n)\times m} \end{pmatrix},$$

and recalling the matrix Z in (6), it can be shown that

$$\nabla\phi_{\tilde{\boldsymbol{g}}}(\mathfrak{s}) = 2\sqrt{2m+n}\,P\mathscr{F}^{-1}(\mathscr{F}^{-1}(M\tilde{\boldsymbol{g}}) \times \mathscr{F}(z_{\mathrm{supp}})),$$

$$\nabla\psi_{\mu_1,\mu_2}(\mathfrak{s}) = 2\mu_2\,\mathrm{Re}(Z^{\mathrm{T}}\mathscr{F}^{-1}(z_{\mathrm{reg}})),$$

$$\nabla\chi_{\mu_3}(\mathfrak{s}) = 2\mu_3\left(\sum_{i=1}^{2p+1}\mathfrak{s}_i - 1\right)\boldsymbol{e},$$

$$\nabla\omega_{\mu_4}(\mathfrak{s}) = -2\mu_4(2m+n)^{-1}Z^{\mathrm{T}}\mathscr{F}(\mathscr{F}^{-1}(z_{\mathrm{PSF}}) \div (\mathscr{F}(\boldsymbol{s}) \times \mathscr{F}(\boldsymbol{s}))).$$

One of the local minima of $J_{\tilde{\boldsymbol{g}},\boldsymbol{\mu}}(\mathfrak{s})$ is taken as the inverse filter that provides the best approximation to $\tilde{\boldsymbol{f}}$. In the minimization procedure, at each step of CG we consider the restriction

$$J_{\tilde{\boldsymbol{g}},\boldsymbol{\mu}}|_{\mathfrak{s},\boldsymbol{d}}\colon \mathbb{R} \to \mathbb{R}^+, \quad J_{\tilde{\boldsymbol{g}},\boldsymbol{\mu}}(\alpha)|_{\mathfrak{s},\boldsymbol{d}} = J_{\tilde{\boldsymbol{g}},\boldsymbol{\mu}}(\mathfrak{s} + \alpha\boldsymbol{d}),$$

where $\mathfrak{s}, \boldsymbol{d} \in \mathbb{R}^{2p+1}$; in particular, $\boldsymbol{d}$ represents a descent direction. Fig. 4 shows a case of piecewise convexity of $J_{\tilde{\boldsymbol{g}},\boldsymbol{\mu}}|_{\mathfrak{s},\boldsymbol{d}}$ and thus of $J_{\tilde{\boldsymbol{g}},\boldsymbol{\mu}}$.

CG is able to find the global minimum of $J_{\tilde{\boldsymbol{g}},\boldsymbol{\mu}}(\mathfrak{s})$ only if $J_{\tilde{\boldsymbol{g}},\boldsymbol{\mu}}(\mathfrak{s})$ is strictly convex, but we have seen that imposing constraint (10), while controlling the noise, might prevent the convexity of the cost function.

Numerical results show, however, that the solution found is always acceptable. In practice we have observed that CG is able to move rapidly toward the region of convexity, without getting too close to the points of discontinuity that might occur along a descent direction, taking a step that allows jumping over such discontinuities. This ability is improved by the additional convexity provided by the PSF support constraints (12).

The convergence characteristics of our algorithm are inherited by those of CG. The computational cost of each iteration is of the order $\mathscr{O}[(2m+n)\log(2m+n)]$ arithmetic floating point operations, where $(2m+n)$ is the problem dimension.

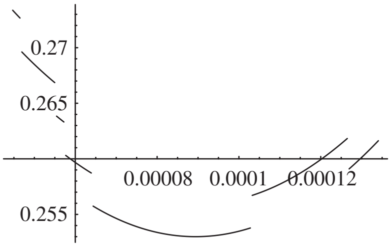

Fig. 4. Example of piecewise convexity of the cost function.

5. Total variation regularization

TV regularization can be used to sharpen an image, increase its contrast and reduce some artifacts that may be due to noise [1,10]. It can be applied, as a final step, to improve the quality of the restored image computed by our algorithm.

Once the *unextended* $\mathfrak{f} \in \mathbb{R}^n$ has been formed from the restored image $\tilde{\boldsymbol{f}} \in \mathbb{R}^{2m+n}$, then the TV regularized image $\boldsymbol{r} = (r_1, \ldots, r_k, \ldots, r_n)^{\mathrm{T}}$ is the solution to the following minimization problem:

$$\min_{\boldsymbol{r}\in\mathbb{R}^n}\left(\frac{1}{2}\|\mathfrak{f}-\boldsymbol{r}\|_2^2+\tau_1\sum_{k=1}^{n-1}\sqrt{(r_k-r_{k+1})^2+\tau_2}\right), \tag{13}$$

where $\tau_1 \geqslant 0$ and $\tau_2 \geqslant 0$ are real parameters; in particular, τ_2 is used to avoid division by zero when computing the gradient of the function being minimized.

TV regularization can be a very effective tool to visually improve the image. Its drawback is the added computational cost involved in solving the nonlinear minimization in (13).

6. Inverse filter estimation

To overcome the fact that the PSF is unknown, the identity operator (with respect to the convolution operator) may be used to form an initial $\boldsymbol{h}$.

In the 2D case, for example, the matrix $\mathfrak{s}$ stores the values of an initial $(2p_y + 1) \times (2p_x + 1)$ null estimate for the inverse filter, created by means of the Dirac delta function:

$$\mathfrak{s} = (\mathfrak{s}_{i,j}), \quad \mathfrak{s}_{i,j} = \begin{cases} 1 & \text{if } i = p_y + 1 \text{ and } j = p_x + 1, \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

In between classic and blind deconvolutions, there also exists the so-called *myopic deconvolution*, in which some estimate of the PSF is available from a previous observation (see for example Section 6.1). Our algorithm can also be viewed as a myopic deconvolution algorithm.

6.1. Astronomical imaging

The atmosphere of the Earth and the instruments of observation both constitute a source of distortion. It is not possible to prepare a reference model, when trying to assess what kind of degradation is in the picture of a distant planet or an object in space. In astronomy applications, though, it is often possible to make use of the a priori PSF information given by a *guide star* (a bright isolated point source).

A guide star $\boldsymbol{g}_{\text{star}} \in \mathbb{R}^{2m+n}$ is a reference image, natural or artificially created, observed under atmospheric circumstances similar to those under which $\boldsymbol{g}$ was observed. The true image $\boldsymbol{f}_{\text{star}}$ of the guide star is a white small point in a black background and its total energy is assumed to be unity. As a consequence, the effect of $\boldsymbol{f}_{\text{star}}$ in the convolution with the PSF can be assumed to be very close to that of the identity operator and thus $\boldsymbol{g}_{\text{star}} \approx \boldsymbol{h}$. In other words, $\boldsymbol{g}_{\text{star}}$ can be used as an estimate of the initial PSF or, equivalently, to form an estimate of the initial inverse filter

$$\boldsymbol{s} = (2m + n)^{-1}\mathscr{F}^{-1}\left(\boldsymbol{e} \div \boldsymbol{\lambda}\right), \tag{15}$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k, \ldots, \lambda_{2m+n})^{\mathrm{T}}$ is defined as

$$\lambda_k = \begin{cases} (\mathscr{F}(\boldsymbol{g}_{\text{star}}))_k & \text{if } |(\mathscr{F}(\boldsymbol{g}_{\text{star}}))_k| > \rho, \\ \mathrm{e}^{-2\pi \mathrm{i} m(k-1)/(2m+n)} & \text{otherwise,} \end{cases} \tag{16}$$

and the role of the real threshold $\rho > 0$ can be interpreted as follows.

The randomly distributed noise affecting $\boldsymbol{g}_{\text{star}}$ may lead to a poor quality inverse filter estimate, yielding large relative perturbations to those spectral components of $\boldsymbol{h}$ that are small (in modulo). By employing $\mathscr{F}(\boldsymbol{g}_{\text{star}})$, a threshold $\rho > 0$ is computed, which corresponds to the value at which the Fourier coefficients stagnate, indicating that the noise starts to dominate.

Only the least perturbed spectral components of $\boldsymbol{g}_{\text{star}}$ are taken into account in estimating the initial inverse filter, while its remaining components are set as in the case in which no guide star is available. In (16) the term $\mathrm{e}^{-2\pi \mathrm{i} m(k-1)/(2m+n)}$ represents the kth Fourier component of an initial estimate defined as in (14).

7. Numerical examples

A package **BDrestoration**, collecting the restoration routine together with auxiliary routines, has been originally developed in the Release 4.1 of *Mathematica* [12] and recently extended to Version 5.1. A multiframe extension and upgrade of the code to Version 5.1 is also under development [13].

All experiments are carried out on a Pentium III 800 MHz processor with 128 Mb of RAM, running Red Hat Linux operating system release 9. Floating point double precision arithmetic is used to obtain approximatively 16 decimal digits of precision.

7.1. Simulated data

Our algorithm is tested here on a blurred and noisy image, obtained from a synthetic 2D image that is first convolved with a Gaussian PSF and then added with Poisson noise. The notation used is the same as for the 1D case, but now variables denote matrices instead of vectors.

The discrete PSF is stored in a $(2m_y + 1) \times (2m_x + 1)$ matrix $\mathfrak{h}$:

$$\mathfrak{h} = (\mathfrak{h}_{k,j}), \quad \mathfrak{h}_{k,j} = \frac{\mathrm{e}^{-[(k-1-m_y)^2 + (j-1-m_x)^2]/c}}{\sum_{a=1}^{2m_y+1} \sum_{b=1}^{2m_x+1} \mathrm{e}^{-[(a-1-m_y)^2 + (b-1-m_x)^2]/c}}. \tag{17}$$

Fig. 5(a) shows the PSF corresponding to values $m_y = m_x = 7$, and $c = 10$.

The original monochromatic image $\mathfrak{f}$ has dimensions $n_y \times n_x$. Here we choose $n_y \times n_x = 15 \times 45$, so that its extension $\boldsymbol{f}$, shown in Fig. 5(b), has dimensions $(2m_y + n_y) \times (2m_x + n_x) = 29 \times 59$.

The image $\mathfrak{f}$ is convolved with the PSF to obtain a blurred image.

Random noise $\boldsymbol{\delta}$ is then added to the convolution, in such a way as to ensure that the components of the simulated observed $\tilde{\boldsymbol{g}}$ remain non-negative. The noise follows a Poisson statistical distribution, with mean β, and is scaled by γ.

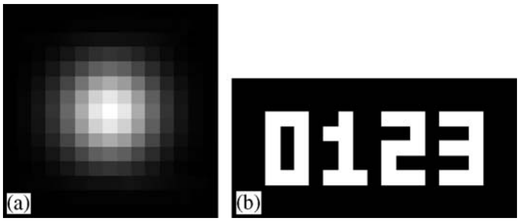

Fig. 5. Simulated data: (a) PSF; (b) Original image.

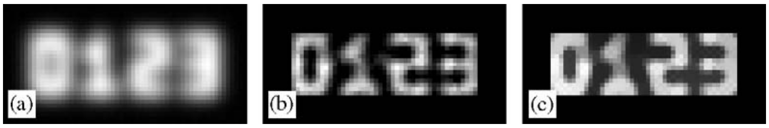

Fig. 6. First test on simulated data: (a) Observed $\tilde{\boldsymbol{g}}$; (b) Restored $\tilde{\boldsymbol{f}}$; (c) Regularized $\boldsymbol{r}$.

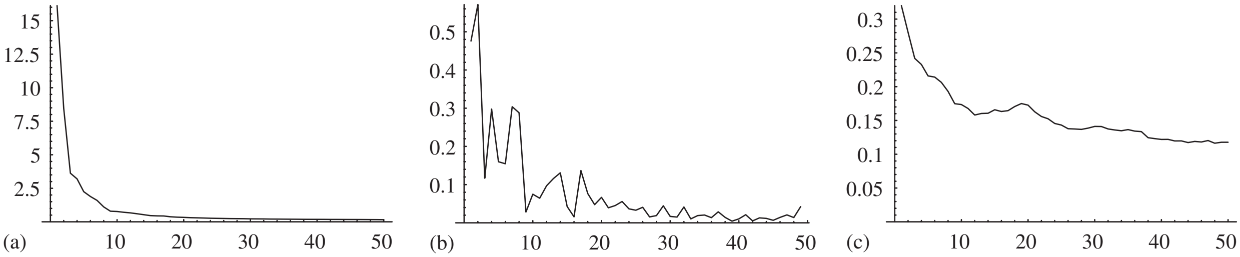

Fig. 7. Behavior of the algorithm during the iterations (test on simulated data): (a) Cost function; (b) Relative variation; (c) Relative error.

A common way to measure noise is by the *signal to noise ratio* expressed in decibels (dB):

$$\mathrm{SNR} \overset{\mathrm{def}}{=} 20 \log_{10} \sqrt{\frac{\sum_{k=1}^{2m_y+n_y} \sum_{j=1}^{2m_x+n_x} g_{k,j}^2}{\sum_{k=1}^{2m_y+n_y} \sum_{j=1}^{2m_x+n_x} \delta_{k,j}^2}}. \tag{18}$$

As the noise increases, the SNR decreases. Setting the noise parameters to the values $\beta = 5$ and $\gamma = 10^{-3}$ results in a SNR of approximately 35 dB, which affects the $\tilde{\boldsymbol{g}}$ shown in Fig. 6(a).

An initial estimate of the support of the inverse filter is needed, to form the initial matrix $\mathfrak{s}$ of dimensions $(2p_y + 1) \times (2p_x + 1)$ and created as in (14). Here we choose $p_x = p_y = 10$.

We now apply the restoration algorithm to $\mathfrak{s}$ and $\tilde{\boldsymbol{g}}$. In this first test the exact image support $n_y \times n_x = 15 \times 45$ is employed. The regularization parameters are set to the values $\mu_1 = 5$, $\mu_2 = 1$ and $\mu_3 = \mu_4 = 1/10$. Fig. 6(b) shows the restored image obtained after 50 iterations; the computational time is 69 s.

As a final optional step, TV regularization can be applied to the restoration to obtain a sharper image: the TV parameters are set to the values $\tau_1 = \frac{1}{8}$ and $\tau_2 = 10^{-5}$. The time for this last step is 19 s. The final result is shown in Fig. 6(c).

7.2. Implementation details

A 'log' information file is available that is useful to analyze the behavior of the algorithm throughout the iterations (see Fig. 7). Our algorithm returns:

- the value of the cost function $J_{\tilde{\boldsymbol{g}},\boldsymbol{\mu}}(\mathfrak{s})$ at each iteration;
- the relative variation between the values of $J_{\tilde{\boldsymbol{g}},\boldsymbol{\mu}}(\mathfrak{s})$ at two consecutive iterations;
- the relative error between restored and true image, which is available here as we are considering simulated data.

The stopping criterion is based on a maximum number of iterations and on default tolerances on both the value of $J_{\tilde{\boldsymbol{g}},\boldsymbol{\mu}}(\mathfrak{s})$ and its relative variance. Default values are 100 iterations and tolerances of 10^{-6}, but they can be set by the user.

7.3. Comparison with NAS–RIF and NSR–RIF

In this section we summarize the results of comparing our algorithm with two other methods, namely NAS–RIF [9] and NSR–RIF [10], that are well known in the field of blind deconvolution and from which our method stems.

Table 1
Error and timing results on the simulated data, for fixed $SNR = 35$ and varying values of $\boldsymbol{\mu}$

$\boldsymbol{\mu}$	NAS err	NSR err	NSH err	NAS time	NSR time	NSH time
(1/50, 1, 1/10, 1/10)	0.133914	0.136902	0.119281	48.29	67.59	92.94
(5, 1, 1/10, 1/10)	0.133914	0.133914	0.116076	48.46	57.07	75.57
(3/2, 120, 1/20, 1/20)	0.133783	0.133783	0.114359	47.89	57.30	71.05
(5, 1/15, 1, 1)	0.156673	0.156673	0.129859	44.00	52.19	74.32

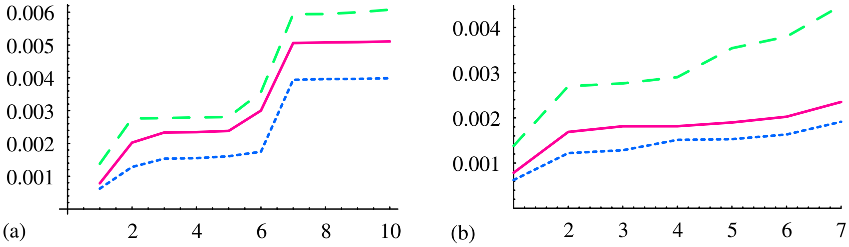

Fig. 8. Work per unit time needed by methods NAS–RIF (split line), NSR–RIF (continuous line) and NSH–RIF (dotted line) to reach an acceptable approximation to a minimum: (a) Fixed SNR, varying $\boldsymbol{\mu}$; (b) Fixed $\boldsymbol{\mu}$, varying SNR.

Table 2
Error and timing results on simulated data, for fixed $\boldsymbol{\mu} = (1/50, 1, 1/10, 1/10)^{\mathrm{T}}$ and varying SNR

SNR	NAS err	NSR err	NSH err	NAS time	NSR time	NSH time
46	0.129514	0.153257	0.126122	47.96	65.14	83.38
37	0.135149	0.140971	0.115590	46.58	77.65	94.76
35	0.133914	0.136902	0.119281	48.46	67.59	92.94
25	0.174132	0.174021	0.173740	45.86	95.90	90.79
22	0.168675	0.168381	0.166636	47.62	99.70	102.11
16	0.204495	0.204845	0.201857	45.60	107.90	132.03

The problem setting is that described in Section 7.1, related to the simulated data. For the fixed value 35 dB of the SNR, we examine the behavior of the three methods of interest, varying the regularization parameters as shown in Table 1.

While performing in a way similar to that of NAS–RIF and NSR–RIF, with respect to the quality and locality of convergence, our algorithm is almost always able to reach a smaller error value. The price to pay for this improvement is the higher computational cost. By measuring the work per unit time that is necessary to reach an acceptable approximation to a minimum, though, we can show that this cost is lowest for our method. The behavior shown in Fig. 8(a) is computed for the results of Table 1, but it holds for all the numerical tests performed.

We now carry out another comparative experiment. The problem setting is, again, that of the simulated data in Section 7.1; here, though, we fix a quadruple of regularization parameters and study the behavior of the three restoration methods of interest, as the SNR varies.

Let $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3, \mu_4)^{\mathrm{T}} = (1/50, 1, 1/10, 1/10)^{\mathrm{T}}$ and let the SNR assume different values corresponding to different levels of noise affecting the observed image. Table 2 confirms that our method reaches a smaller error value than NAS–RIF and NSR–RIF, but it requires a higher computational time. As before, the work per unit time is lowest for our method. The behavior shown in Fig. 8(b) is computed with the results of Table 2, but it holds for all the numerical tests performed.

Perhaps more interesting is to observe the convergence quality of the three methods. As the SNR decreases in Table 2, the problem solved by the three restoration methods has increasing complexity and worsening conditioning.

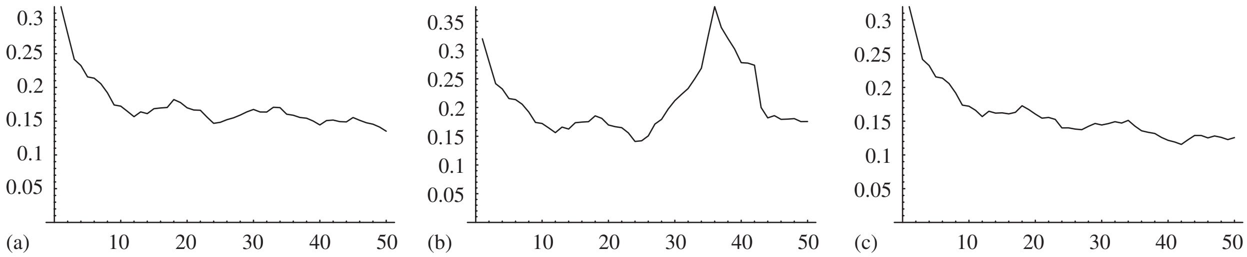

Fig. 9. Error behavior on the simulated data, with SNR $= 37$ and $\boldsymbol{\mu} = (\frac{1}{50}, 1, \frac{1}{10}, \frac{1}{10})^{\mathrm{T}}$: (a) NAS–RIF; (b) NSR–RIF; (c) NSH–RIF.

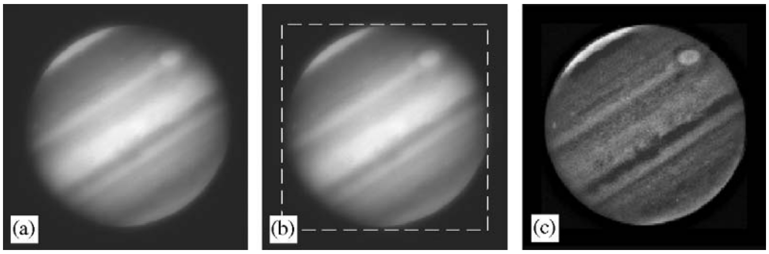

Fig. 10. Images of Jupiter: (a) Observed image; (b) Support estimation; (c) Restored image.

This situation does not affect the time taken by NAS–RIF to compute a solution. We recall that for this method the conditions for the convexity of the cost function are satisfied.

The comparison, here, is therefore mostly between NSR–RIF and our method, since both of them do not satisfy the convexity conditions. By introducing the constraints (12) on the PSF support, our algorithm adds further convexity to the cost function and in this way improves the convergence of the CG iteration. Fig. 9 illustrates how the better convexity features of our algorithm, over those of NSR–RIF, can help in avoiding discontinuities.

The quality of convergence remains obviously local, but the extra constraint allows our method to generally reach a somewhat smaller error than NAS–RIF and NSR–RIF, as illustrated by the error results in Tables 1 and 2.

7.4. Real astronomical data

As a more realistic example, we consider a real life image of Jupiter in Fig. 10(a): this 512 × 512 image is stored in the matrix $\tilde{\boldsymbol{g}}$. The support values $n_x = 430$ and $n_y = 428$ of the desired original image are chosen heuristically, with the visual aid obtained by plotting a dashed white rectangle (the estimated support region) around $\tilde{\boldsymbol{g}}$, as illustrated in Fig. 10(b).

An estimate of the rectangular support of $\mathfrak{s}$ is used ($p_x = p_y = 17$) to define the initial inverse filter as in (14). The minimization parameters are $\mu_1 = 10$, $\mu_2 = 5$, $\mu_3 = 10$ and $\mu_4 = 1$. The restored image after three iterations is shown in Fig. 10(c); the time taken is 537 s.

7.5. Guide star

The aim of this experiment is to test the effectiveness of the restoration when an inverse filter estimate is available by means of a guide star, as described in Section 6.1. In this case a PSF is available in the form of a real life data image; it has dimensions 255 × 255, that are also the dimensions of all the images occurring in this example.

The reference image for the guide star is rendered as a white point in a black background; its observed image is simulated by the convolution with the given PSF and by the addition of approximately 25 dB of noise (following a Poisson distribution, with parameter $\beta = 5$, and scaled by $\gamma = 10^{-6}/5$). This observed image of the guide star is used to form an initial estimate for the inverse filter, using (15) and (16).

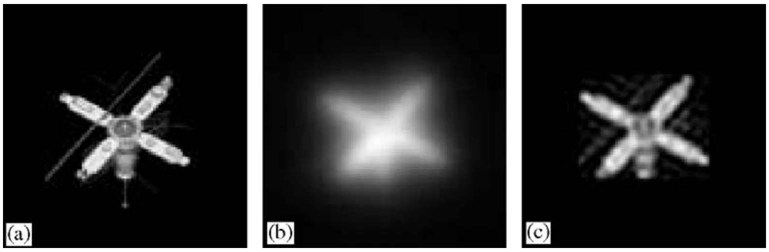

Fig. 11. Satellite image: (a) True image; (b) Observed Image. (c) Restored image.

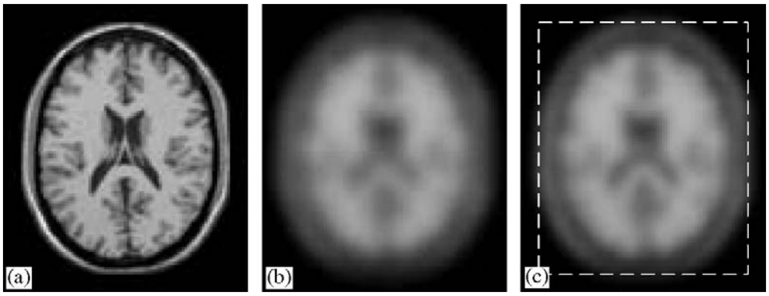

Fig. 12. Brain MRI image: (a) Original image; (b) Observed image; (c) Support estimate.

The reference image for the satellite is a real life data image, shown in Fig. 11(a). Its observed image is simulated by the convolution with the given PSF and by the addition of approximately 25 dB of noise (following a Poisson distribution, with parameter $\beta=4$, and scaled by $\gamma=10^{-3}$). The result is shown in Fig. 11(b). The estimated rectangular support 137×111 of the image is given as input. The minimization parameters are $\mu_1 = \frac{3}{2}$, $\mu_2 = 120$, $\mu_3 = \frac{1}{20}$ and $\mu_4 = \frac{1}{20}$. The restored image after five iterations is shown in Fig. 11(c); the time taken is 250 s.

7.6. Medical data

Magnetic resonance imaging (MRI) is an imaging technique used to produce high quality images of the inside of the human body; it allows the imaging of soft tissue and the metabolic processes therein, providing both morphologic and functional information on the body (in particular, the brain) [5,7]. MRI stems from the application of nuclear magnetic resonance (NMR) to radiological imaging: it is based on the principles of NMR, a spectroscopic technique used by scientists to obtain microscopic chemical and physical information about molecules. In MRI images the PSF is unknown.

We simulate a real situation, applying our algorithm to the blurred and noisy image $\tilde{\boldsymbol{g}}$ of dimensions 100×83, depicted in Fig. 12(b); the original image is a gray-scale image of a human brain, as shown in Fig. 12(a).

The blur is simulated via a PSF defined as in (17) with $m_x = m_y = 7, c = 25$. The Poisson noise has parameter $\beta = 20$ and is scaled by $\gamma = \frac{1}{10}$, approximatively yielding an SNR of 33 dB. The support $n_y \times n_x = 80 \times 70$ is visually estimated from the degraded image and shown in Fig. 12(c).

The restoration algorithm is started on the degraded image: the vector $\boldsymbol{\mu}$ of the regularization parameters is set to $(1, \frac{1}{10}, 10, 1)^{\mathrm{T}}$ and use, as estimated inverse filter, a 15×15 matrix created by means of the Dirac delta function: the elements of this matrix are 1 in position (7, 7) and 0 elsewhere. The image obtained after six iteration, together with its TV regularization, is shown in Fig. 13.

Fig. 14 shows that few iterations are sufficient to obtain a reconstruction that is acceptable in the presence of high noise. The behavior shown in Fig. 14 occurs quite often in real applications and in high dimension problems. Early truncation of the restoring procedure, in fact, can be used as an additional regularization method.

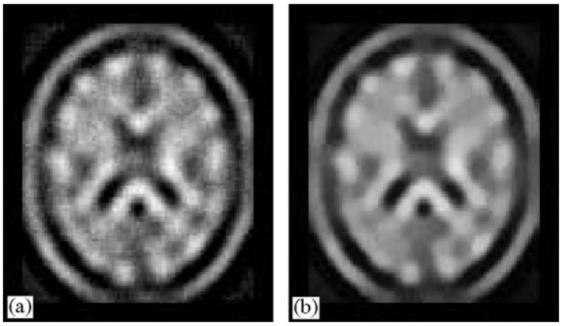

Fig. 13. Results of the restoration of the brain MRI image: (a) Sixth iteration; (b) TV regularization.

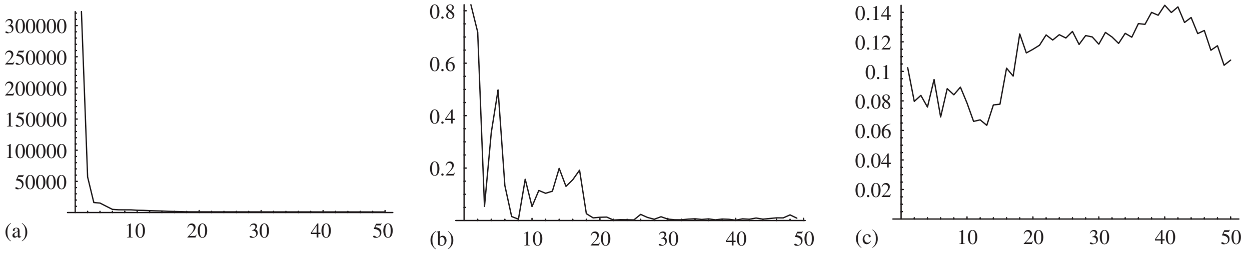

Fig. 14. Behavior of the algorithm during the iterations on MRI data: (a) Cost function; (b) Relative variation; (c) Relative error.

8. Conclusions

The NSH–RIF blind deconvolution algorithm is outlined. A prototype for the restoration of images is implemented in *Mathematica* [14], with which it is possible to construct simulated data for numerical testing as well as processing real data images.

Numerical experiments show the effectiveness of the algorithm in returning good quality images. Images can have high dimensions, leading to a large number of unknowns, given for example by the color of the pixels. The computing time required by the NSH–RIF procedure obviously depends on the dimensions of the target image, but are always acceptable, being of the order of a few minutes in the case of real images.

A comparison with two other methods, namely NAS–RIF [9] and NSR–RIF [10], has been given. While performing in a way similar to that of the two methods mentioned, our algorithm is generally almost always able to reach a smaller error. By introducing a constraint on the PSF support, our algorithm adds further convexity to the cost function and in this way improves the convergence.

Sometimes $k > 1$ different images $\{\tilde{\boldsymbol{g}}^{(1)}, \ldots, \tilde{\boldsymbol{g}}^{(k)}\}$ of the same object $\boldsymbol{f}$, subject to different noises $\{\boldsymbol{\delta}^{(1)}, \ldots, \boldsymbol{\delta}^{(k)}\}$, are available. In such a case a *multiframe* extension of the NSH–RIF should be considered and is currently under development [13]. Preliminary tests indicate that this approach is effective when the noise is high, since it prevents noise amplification.

The use of a globally convergent procedure for the minimization of the cost function will also be evaluated in the future: the effectiveness of quasi-Newton methods for this kind of optimization problem is under investigation.

Acknowledgements

We are grateful to Prof. Reichel, for his suggestions and improvements to this paper, and to Dr. Sofroniou, for his help with the latest version of the package and with this research. We further wish to thank Proff. Sergeyev and Sordoni for helpful discussions. This work has been partially supported by the GNCS Research Project 2002 'Analysis and Development of Parallel Computational Kernels for Problem Solution'.

References

[1] M. Bertero, P. Boccacci, Introduction to Inverse Problems in Imaging, Institute of Physics Publishing, Philadelphia, PA, USA, 1998.

[2] D. Calvetti, B. Lewis, L. Reichel, Smooth or abrupt: a comparison of regularization methods, in: F.T. Luk (Ed.), Advanced Signal Processing Algorithms, Architectures and Implementations VIII, Proceedings of SPIE, vol. 3461, 1998, pp. 286–295.

[3] D. Calvetti, L. Reichel, Q. Zhang, Iterative solution methods for large linear discrete ill-posed problems, Appl. Comput. Control, Signals Circuits 1 (1999) 313–367.

[4] R. Gonzalez, P. Wintz, Digital Image Processing, Applied Mathematics and Computation Series, Addison Wesley, Reading, MA, USA, 1977.

[5] E.M. Haacke, R.W. Brown, M.R. Thompson, R. Venkatesan, Magnetic Resonance Imaging, Physical Principles and Sequence Design, Wiley–Liss, New York, NY, USA, 1999.

[6] J.B. Hiriart Urruty, C. Lemarèchal, Convex Analysis and Minimization Algorithms 1: Fundamentals, Grundlehren der Mathematischen Wissenschaften Series, Springer, Berlin, Germany, 1993.

[7] J.P. Hornak The Basics of Magnetic Resonance Imaging, 2003, ⟨www.cis.rit.edu/htbooks/mri/⟩

[8] D. Kundur, Blind deconvolution of still images using recursive inverse filtering, Master's Thesis, Electrical Engineering Department, Toronto University, Ontario, Canada, 1995.

[9] D. Kundur, D. Hatzinakos, A novel blind deconvolution scheme for image restoration using recursive filtering, IEEE Trans. Signal Process. 46 (2) (1998) 375–390.

[10] M. Ng, R.J. Plemmons, S. Qiao, Regularization of RIF blind image deconvolution, IEEE Trans. Image Process. 9 (6) (2000) 1130–1134.

[11] J. Nocedal, S.J. Wright, Numerical Optimization, Operations Research Series, Springer, Berlin, Germany, 1999.

[12] G. Spaletta, L. Caucci, Blind Restoration of Astronomical Images, 2005, MathSource, ⟨library.wolfram.com/⟩

[13] G. Spaletta, Parallel multiframe blind restoration, 2005, preprint.

[14] S. Wolfram, The Mathematica Book, fifth ed., Wolfram Media, Champaign, IL, USA, 2003.